AD-A115 538 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 5/1
A MANAGEMENT SYSTEM FOR COMPUTER PERFORMANCE EVALUATION.(U)
DEC 81 H K BIRCH
UNCLASSIFIED AFIT/GCS/EE/81D-1 NL

1 OF 2
A
A115538

DDC

1

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY (ATC)

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

1

A MANAGEMENT SYSTEM FOR COMPUTER PERFORMANCE EVALUATION

AFIT/GCS/EE/81D-1

Harry K. Birch
Captain USAF

AFIT/GCS/EE/81D-1

A MANAGEMENT SYSTEM FOR COMPUTER PERFORMANCE EVALUATION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Harry K. Birch
Captain USAF

December 1981

Approved for public release; distribution unlimited.

PREFACE

As an installation manager of a Burroughs 3500 I encountered many problems concerning its performance. These problems ranged from customers complaining about slow turnaround time to the impact of having to add additional workload on a seemingly overloaded system. In dealing with these problems, I learned first hand the importance and difficulties of a computer performance evaluation (CPE). The major difficulties I encountered with CPE were when should I start a performance evaluation, what areas I should study, what CPE tool or techniques to select, and finally, how do I organize the effort. As a manager I felt that I needed a reference or tool that would broaden my CPE knowledge and assist me in answering these questions. Regretfully, I had no such tool or reference and I was forced to rely on my own minimal knowledge and experience. This is why I decided upon this subject for a thesis investigation.

Installation managers faced with performance problems often make incorrect decisions because of insufficient information. The result of these decisions has been an untold waste of money and resources. In order to make correct decisions concerning performance problems, managers need information that can only be provided by measuring and evaluating the performance of their computer. Since installation managers are often required to make performance decisions, this information is needed continuously. Therefore, installation managers need to develop a comprehensive CPE program or system to insure they receive this important performance information. This CPE program or system can be used by installation managers as a management tool that will assist them in making correct performance decisions. This thesis effort develops such a management tool.

Finally, I would like to take this opportunity to thank those people who made this thesis effort possible. I thank Captain Steve Christiana for suggesting the topic and for the recommendations and support he provided. I thank Dr. Gary B. Lamont for being my advisor and for the guidance and recommendations he provided. Lastly, I thank my wife, Faye, for taking on the difficult task of typist and especially for the spiritual and loving support she provided throughout this endeavor.

Contents

List of Figures

List of Tables

Abstract

This study discusses the design and implementation of a management system that will provide an installation manager or manager of a computer system with the means to measure and evaluate the performance of their computer system. This system is composed of three parts; information, people, and reports. The information part of this system is a set of factors that can cause problems with computer performance and the data which can be gathered by various CPE tools and techniques used to solve these problems. The factors and data of the information portion of this management system are presented and discussed in this paper.

The people part of this system are members of a CPE team. They are individuals familiar with the organization, the workload, and the computer system hard~~ware~~/software. It is a team that can either use or learn to use the tools and techniques of computer performance evaluation. The make-up of such a team is also discussed in this paper.

The reports section of this system is the most important part because this is what the installation manager or computer system manager will use to determine the performance of their computer system. The responsibility of the reports and their accuracy lies with the CPE team. This paper disucsses some of the reports that a CPE team can generate.

Also included in this study is background information on computer performance analysis as well as explanations and

definitions of many of the CPE tools and techniques used by CPE analyst. This study omits much of the technical jargon associated with CPE; however, references are provided for those wishing a deeper understanding.

This study was conducted at the request of the Systems Engineering Avionics Facility of the Aeronautical Systems Division and as such, the implementation and recommendation parts of this paper relate solely to them. Although this study was conducted for a specific orgnization, the management system presented in this document could be used at any computer installation or data center.

I. Introduction

The computer (information processing) industry is now the second largest industry in the world, second only to energy and is forecast to reach "first place" in the 1980's. Still, with so large an industry and so large an investment in systems, equipment, and specialized people, relatively little management attention has been paid to the efficiency with which this industry operates and whether it can operate more productively and effectively. (Ref. 6: II-1.)

The issue of operating more productively and effectively has concerned users throughout the history of computer evolution. As such, a term was created to identify the effectiveness and efficiency of computers. This term is called computer performance evaluation (CPE).

Broadly defined, computer performance evaluation deals with the methods used to collect information that reflects the computer system performance, analysis tools, and techniques used to evaluate this data and the formulation of policy necessary to bring the performance of the computer system in line with operational goals. (Ref. 18 : 7)

The General Accounting Office estimates that the utilization of federal computer systems could be improved from 20-40% with the aid of a computer performance evaluation. (Ref. 17 : VI-57) To be conservative, let us say that the utilization can be improved by 25%. This means that if a computer system takes twenty-four hours to service all its customers; by performing a computer performance evaluation, the same computer system could service these users in 18 hours. This would be a tremendous benefit for organizations that continuously face backlogs and never finish processing. If it is possible to improve the utilization of computer systems by as much as 25%, why are not all computer installations operating at this improved level? The problem is that installation managers and managers of computer systems do not thoroughly understand CPE, nor do they know how to start and continue a CPE effort or what direction to take.

Background

The growing complexity of computer systems has been accompanied by the growing need for more information about what is actually taking place inside and outside these systems. As a result, the interest in the field of computer performance evaluation has grown tremendously.

First generation computers (vacumn tubes) were designed in the early and mid 1950's to process as fast as possible in two principal areas of application; scientific and commercial. The scientific processors were judged by how fast they could add, subtract, multiply, and divide; the commercial processors were judged by how fast they could manipulate data. These early processors were organized to operate serially, that is, they had to input the program and data before processing could begin. While processing the computer could do no I/O. Upon completion of processing the information was output. Evaluating the performance of these early computers was easy and the only tool required was a stopwatch. (Ref. 14 : 1-1, 1-2)

During the late 1950's and early 1960's second generation computers (solid state) emerged. Also, with the development of larger and less expensive memories, various software aids such as assemblers and compilers began to play an important role in the performance of computers. Because of their faster computational capability, faster input-output devices became standard equipment. These systems were more productive because they were able to execute program instructions and perform input and output functions simultaneously. This was accomplished by a new type of computer program called an operating system which provided for transition from one computer program to another and for control over input-output procedures. During this second generation era, evaluating computer performance became more difficult. No longer could managers use a stop watch approach to measure the performance of their computer system. (Ref. 20 : VI-8) Unfortunately, evaluating the performance of these computers received little attention from managers who often had only an elementary understanding of automatic data processing (ADP) operations and were not in a position to have much impact on insuring efficient ADP operations. Very few tools were developed and very little was written about computer performance evaluation for the problem of measuring the performance of computer systems was just beginning.

Third generation computers emerged in the mid 1960's and were smaller in size but normally able to compute and process data much faster. They were modularly designed so that their capacities could be increased as an organization's data processing needs increased. Operating system software became more complex because it now controlled several computer programs which operated concurrently in the computer system (multiprogramming). With this advanced hardware and operating systems, also, came continued growth in applications. Industry and business began to rely more and more on computers and a tarememdous growth in the computer industry began. In addition to batch processing, new applications with characteristics of remote access, online processing, and real time processing were developed. Third generation computers were more technical and ADP managers faced more difficult and demanding tasks in attempting to improve the efficiency and effectiveness of their computer systems. Cost began to plan an important role in the life of a computer system and installation managers were forced to become aware of the effectiveness and efficiency with which these computer systems were operating. As a result, computer performance evaluation received a new precedence. (Ref. 20: VI - 8,9)

From the emergence of third generation computers to now, much has been written about computer performance evaluation. Countless articles, pamphlets, special studies, and books have been published on the subject. In addition, organizations such as the Computer

Performance Evaluation Users Group and Computer Performance Measurement Group have been formed and hold annual meetings where they present papers and discuss matters related to computer performance evaluation. These organizations are composed of CPE analyst from both the government and civilian sectors who are experienced in the field of CPE. Coupled with this increased development of tools and techniques was an increased awareness of the importance of CPE by installation managers and system managers. Although much has been written on CPE and many tools and techniques have been developed to assist the computer analyst, little has been done to assist the manager of the computer installation. Today, a manager of a computer installation faced with a performance problem must rely on his own knowledge and experience to derive a solution. Oftentimes, the manager knows what information is needed to solve the problem but does not know how to obtain it. Other times the manager may not know where to start. Since the field of computer performance evaluation is expanding along with the capability and complexity of new computers, installation managers are faced with several very difficult problems. The first of these problems is how to measure and evaluate the performance of present day computer systems and secondly, where does a computer performance evaluation begin.

Problem Statement

Managers of computer installations and computer systems need help when measuring and evaluating the performance of their computer systems. This process of measuring and evaluating the performance of computers and computer systems has become important, demanding, and difficult. It is important because performance is one of the prime considerations used by managers when evaluating a computer or computer system. This process is demanding because it requires a thorough understanding of the levels of a computer system and an understanding of the user's habits, preferences, and adaptability to system changes. It is difficult because of the complexity of present day computer systems and the fact that these systems exhibit different characteristics and one method of analyzing the same system characteristic might not be applicable under both circumstances. This process is not only important, demanding, and difficult; it also requires the manager to answer some important questions. These questions are:

- How and where do you start an evaluation?
- Does the whole system need evaluating or just elements of it?
- What tools and techniques can be used and what are the advantages and disadvantages of each?
- How are these tools and techniques selected and what is the cost?
- Once a tool or technique is obtained, how long will it take to provide results ?
- How difficult are these tools and techniques to use and do they require the hiring of additional personnel?

The answers to these questions are important because they determine how and by what means the perforamnce of a computer or computer system can be measured.

Because computer performance evaluation is so demanding and difficult, many managers are incapable of answering these questions. When confronted with computer performance problems, these managers sidestep the issue of computer performance evaluation and rush out and purchase hardware they do not need in an attempt to come up with a quick-fix or they spend thousands of dollars on computer performance analysts who discover the problem but only after spending considerable time, money, and effort. This time, effort, and money need not be spent if the manager had a system that could provide information about the performance of the computer system. This information could allow the manager to make a quicker, less expensive, and more accurate decision to solve the problem.

A system that can help a manager measure and evaluate the performance of a computer system is needed by all managers of computer systems and computer installations. These managers are continually faced with problems concerning computer performance which they cannot solve themselves because they do not have the knowledge nor experience needed to deal with the complex issues of computer performance evaluation. Neither, do they have the ability to answer the questions on the previous page.

Scope

This thesis develops a management system that can help managers of computer installations and computer systems obtain performance information about their computer systems. It also provides managers with the means to identify problems before they occur and to plan for future computer resource needs.

This thesis presents and discusses the aspects of a computer facility that can cause poor performance. Discussed in detail are aspects of the organization, the workload, and the computer hardware/software. Also discussed is how to establish a computer performance evaluation team. This team is the most essential part of the management system because it frees the manager from having to deal with complex computer performance evaluation problems and places this responsibility on a group of individuals more knowledgeable and capable of dealing with these problems. The major products produced by the CPE team are reports which provide the manager with the means to measure the performance of the computer system. These reports also provide the manager with information that can be used to identify performance problems before they occur and to plan for future needs of computer resources. The reports presented and discussed in this thesis are only the major ones since there are many different kinds and types of CPE reports that can be generated. The format of these reports is briefly discussed and is left for the managers and members of CPE teams to determine.

The management system presented in this thesis can be implemented at any computer installation or data center; however, for this thesis the implementation and recommendations sections pertain only to the Systems Engineering Avionics Facility, the sponsor for this thesis effort.

Approach

The basic requirement for the development of this management system is to determine what kinds and types of CPE information managers of computer systems and computer installations such as SEAFAC need. Some of this information was obtained from my own experience as an installation manager and from discussions with Captain Steve Christiani, the computer system manager of SEAFAC. The remainder of this information came from an extensive literature search in the areas of computer performance evaluation and measurement. The books, reports, and documents reviewed and studied to obtain this information are presented in the bibliography section of this paper. The information obtained from my personal experience and discussions coupled with the information obtained from the study of CPE related books, documents, and reports provided the kinds and types of CPE information needed by managers. Once this information was determined, the next step was to determine how to gather it. Fortunately, most of this information was found in articles and text books; however, some of the information needed was about the organization, the service it provides, and how it provides that service. Therefore, to obtain this information, an analysis approach was taken.

The first step in this analysis is to gather information on the organization. The next step is to develop an understanding of the workload. This information provides an insight into some computer system requirements and constraints. Following this is the most difficult task; becoming familiar with the specific computer system's hardware/software.

By combining the information from the research, analysis, and discussions, with my experience, the computer performance evaluation management system was developed.

Order of Presentation

Chapter II discusses the requirements of a CPE management system. Knowledge needed about the organization, the workload, and computer system is presented, along with diagrams and questionaires to assist in obtaining this knowledge. Chapter III describes the design of a CPE management system. Included in this chapter are the objectives of the management system and the measures and reports that the system can collect and present. Chapter IV and V cover the implementation and recommendations of this plan for the Systems Engineering Avionics Facility of ASD. The conclusion of the thesis is presented in Chapter VI.

II. The Requirements of a CPE Management System

The requirements of a CPE management system are few. First, a team of specialized people is needed to identify performance problems and recommend solutions. Second, is a series of reports and measures tailored to meet management needs, and lastly, is information to assist the members of the team identify these problems and recommend accurate solutions. The people and report requirements of this management system will be presented later. This chapter focuses on the information requirement. Since all aspects of a computer center or installation either directly or indirectly impact performance, information about all aspects of the computer center or installation must be obtained. To obtain this information, the computer center or installation is divided into the organization, the workload the organization processes, and the computer system the organization uses to process this workload.

This chapter presents and discusses the factors of a computer center or data center that can cause poor performance, as well as how to find them. Specifically, the following areas will be discussed along with their interactions:

- Organization
- Workload
- Computer System

Since a computer system is normally composed of two integrated systems, hardware and software; these systems will be discussed separately. Figure 1 is a data flow diagram of How To Get Started.

Since this thesis uses data flow diagrams, a brief definition

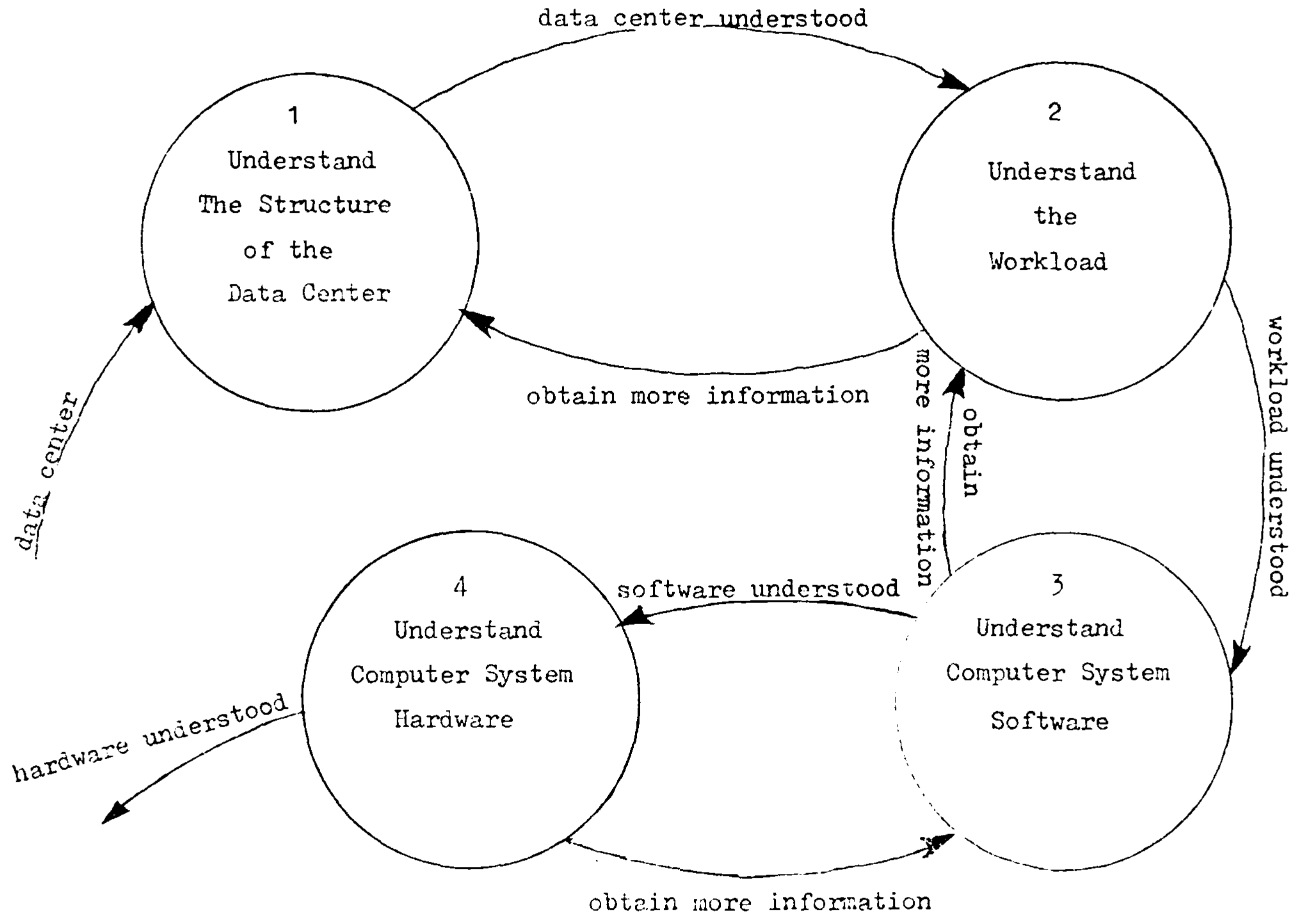


Figure 1 Data Flow Diagram of How To Get Started

is given for those who are unfamiliar with the term. A data flow diagram is a graphic tool that represents data flow and transforms in a process. It can be used in a systems development environment to emphasize the logical flow of data in a system, while deemphasizing procedural aspects of the problem and physical solutions. The basic symbols of a data flow diagram are called transforms; these are represented by circles, each identifying a function that transforms data. The circles are connected by labeled arrows which represent the inputs to and the outputs from the transforms. Each transform is numbered and can be expanded to show more detail. (Ref. 15: 68)

Organization

A thorough understanding of the computer installation is required to effectively develop a computer performance evaluation management system. Specifically, the situation or circumstances that provoked the performance effort should be found along with the position of the computer installation with respect to the organization that it serves. The computer installation's operational objectives must be obtained and understood. An organization chart should be acquired and studied to determine how the manager's programmers, operators, etc. are structured within the computer center. The procedures used by management to govern the computer center should be obtained and studied. In addition, the number of hours the computer center is operational should be obtained along with any scheduled or unscheduled closings, such as on holidays or during severe weather. This information provides the foundation from which more specific and detailed information about the organization can be added. (Ref. 1: 11)

Information on the number of programmers, operators, maintenance technicians, and computer system analysis should be obtained as well as the hours each works. In addition, information should be obtained about priorities and schedules and the impact they have on processing, assuming they exist in the organization. Information should be gathered on the critical jobs the computer center processes such as payrolls or monthly reports. The information gathered so far will assist the analysts in understanding the service the organization provides and how it provides that service.

Perhaps the most important information that must be gathered is that from the managers. Since the program is being developed primarily for management, the kinds and types of information they need to make performance decisions should be obtained directly from them. Additionally, try to determine the way they would like this information presented. The major difficulty with this is that oftentimes management is not totally aware of the information they need to assist them in making computer performance decisions, nor do they know how this information should be presented. Additionally, many managers have not acquired an understanding of computer performance evaluation. This is where the CPE team can begin educating management about CPE; its benefits and advantages as well as assist them in determining the kinds of information they need and how it should be presented. Figure 2 is a data flow diagram depicting this approach.

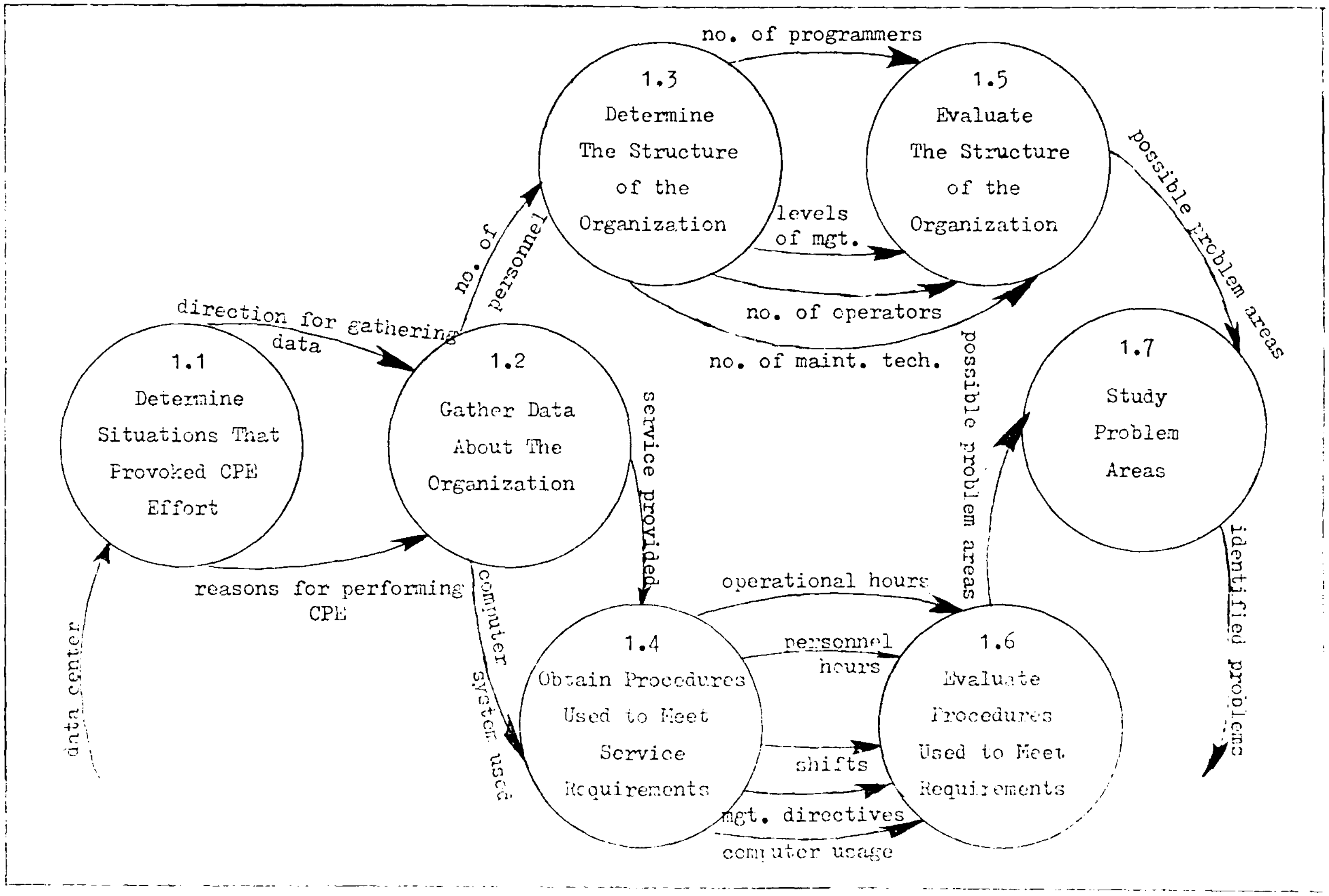


Figure 2 Data Flow Diagram to Understand the Organization

Throughout the gathering of all this information, attempt to identify problems with policies and procedures that could have an adverse impact on performance. Appendix A contains a list of questions that should assist the analyst identify these kinds of problems. In addition, conduct discussions with lower level managers, programmers, maintenance technicians, and operators in an attempt to determine their feelings about the organization and the performance problems they may have encountered and identified. The main objective in developing a thorough understanding of the organization is to identify areas that could cause poor performance.

Appendix B contains a list of questions that will assist the analysts of the CPE team to better understand the organization.

Workload

This section presents information about the workload that can cause poor performance. This information can be used by members of the CPE team to identify problems with the workload and to increase their knowledge of it.

The most important aspect in developing a computer performance evaluation management system is to understand the workload. This is because without the workload, the computer system and organization would probably cease to exist. By developing an understanding of the workload, the computer analyst can better determine the computer system requirements and the impact poor performance will have on the users. This does not mean that every process performed or every job that is executed be understood to the fullest. Rather, an understanding should be developed about the "kinds of processes performed" and the types of jobs executed. This information will provide a starting point for the gathering of more specific and detailed data.

Information should be obtained on specific projects, programs, and personnel that use or request service from the computer center; along with the approximate load in terms of the number of jobs submitted by each. Additional information should be obtained about how jobs are classified. The largest user should be identified as well as the smallest. The exact number of users should be obtained as well as an estimate of the number of jobs they submit. This estimate can be over a period of a day, week, or month. All schedules and deadlines associated with the workload, such as run Job X the first of every month, or Job Y must be run at 0900 everyday, should be obtained. Discussions should be made with all users in an attempt to identify any problems or difficulties they may be experiencing with the computer centers, operating procedures, or computer. While talking with the users, ask if any increase or decrease in their workload is planned. This information provides the analyst with the complexity and extensiveness of the workload. (Ref. 1: 13)

The most important aspect to obtain about the workload is if all jobs are being processed. If they are not; is there a backlog, and if so, how large is it. Additionally, find out if this is a regular occurrence or just sporadic. If it is sporadic, attempt to find out when and if a cause is known, determine when and how jobs are submitted for execution, i.e. via cards or terminal and attempt to obtain the percentage of jobs submitted by each. Identify any bottlenecks associated with the workloads such as all jobs arriving at 0900 or large numbers of jobs needing processing at the end of the month. Along these same lines, attempt to determine when the computer center is the busiest and the least busy. Information should be obtained from the operators as to problems they may have identified with the workload. For example, Job X continuously blows up the first time it runs and always has to be run a second time, or Job A runs and everything else stops. Determine which jobs produce hard copies and identify jobs that must be run twice to fulfill hard copy requirements, e.g. organizations requesting seven or more copies. Determine what accounting data the organization gathers and study this material as this will greatly assist in understanding the workload. When gathering information about the workload, attempt to identify problems with the workload that could cause poor performance. Appendix C contains a list of things to look for.

Since the workload is the major factor that affects performance, understanding it becomes extremely important when faced with a computer performance problem or attempting to develop a CPE management system. Figure 3 is a data Flow diagram depicting the procedure used to gather this information and Appendix D contains a list

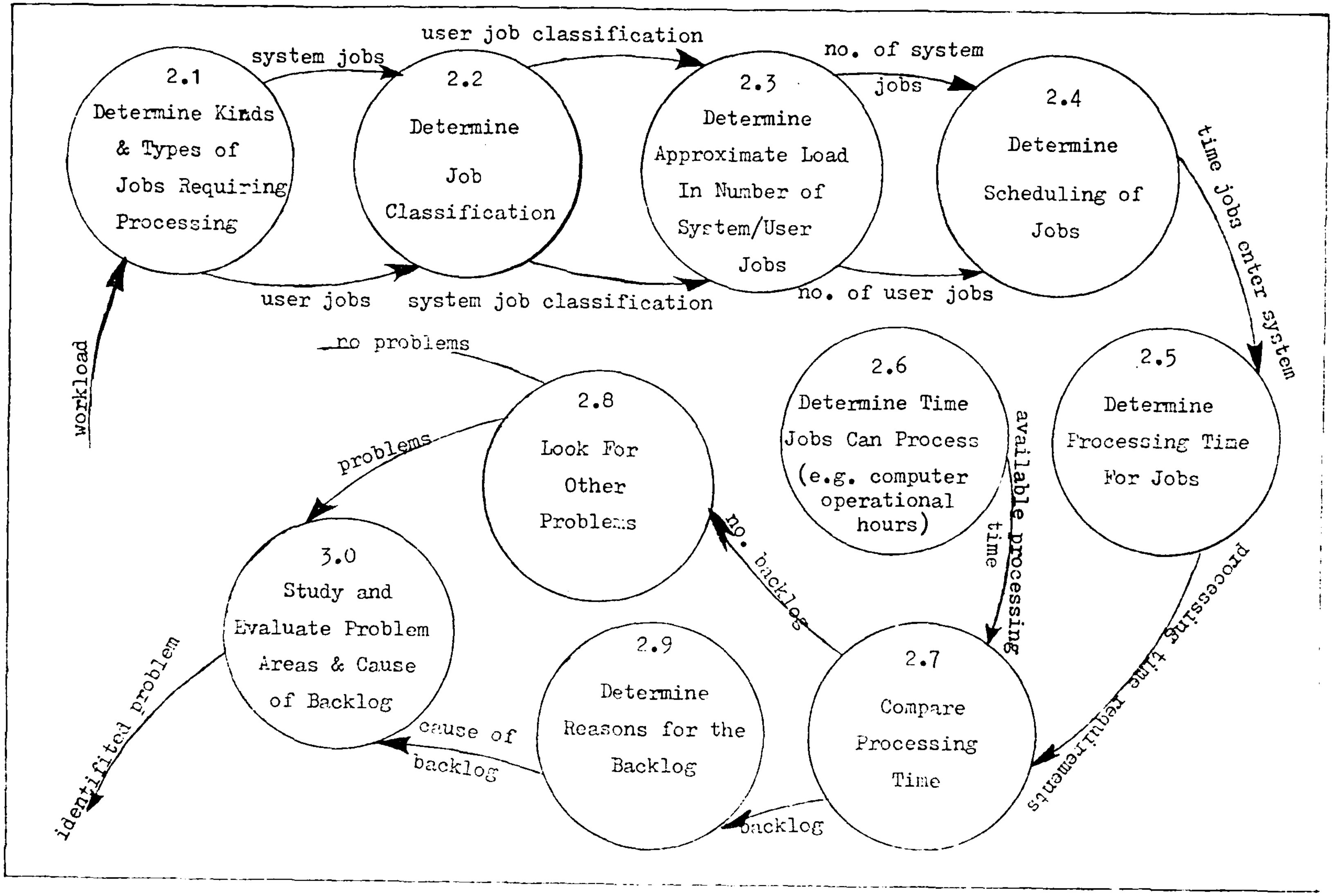

2.1
Determine Kinds & Types of Jobs Requiring Processing
2.2
Determine Job Classification
2.3
Determine Approximate Load In Number of System/User Jobs
2.4
Determine Scheduling of Jobs
2.5
Determine Processing Time For Jobs
2.6
Determine Time Jobs Can Process (e.g. computer operational hours)
2.7
Compare Processing Time
2.8
Look For Other Problems
2.9
Determine Reasons for the Backlog
3.0
Study and Evaluate Problem Areas & Cause of Backlog
workload
system jobs
user jobs
user job classification
system job classification
no. of system jobs
no. of user jobs
time jobs enter system
processing time requirements
available processing time
no. backlog
backlog
no problems
problems
cause of backlog
identifited problem

Figure 3 Data Flow Diagram to Understand the Workload

that will assist the computer analyst to develop a better understanding of the workload.

By using the information in this section, the members of the CPE team, particularly the new ones, should be able to increase their understanding of the workload and to identify workload problems.

Computer System

A computer system is an integrated aggregation of hardware components (central and input-output processors, memories, peripheral devices, interfaces) and software components (the programs which constitute the operating system). (Ref. 4: 3) Since the computer system is the primary tool the organization uses to process its workload, a thorough understanding of how the computer system processes the workload is essential in evaluating the performance of the computer; as well as when developing a system for computer performance evaluation. This does not mean that you, as an installation manager or member of a CPE team, should develop a total understanding of all the little intricasies associated with the computer system such as which registers are used for special purposes or how the system handles interrupts; rather, knowledge should be obtained about the stages a job goes through as it is being processed by the system. As mentioned previously, this paper divides a computer system into two halves, hardware and software. When discussing the hardware/ software parts of a computer system, an important aspect to point out is the level of observation. The level of observation means that different types of people such as managers, operators, programmers, and users view the computer system differently. For example, a manager's view of a computer system is for the most part economic. Maintaining a budget, while satisfying user requirements often is a difficult task for the computer manager. A users view of a computer system is characterized by speed, such as turnaround time or response time. Another user's view could be ease of access and use. These are only a few examples of levels of observation and since this paper discusses a CPE management system to be used by the manager, the members of the CPE team should become familiar with all levels. (Ref. 9: VI 73 - VI 75) The information needed about each half in order to develop a computer performance evaluation system is presented next.

Hardware

Presented in this section is information about computer hardware. This information can be used by members of the CPE team who want to enhance their understanding of computer hardware or identify problems with it.

The first step in understanding the hardware is to determine the make and model of the computer system that the organization uses. The next step is to determine the size of the computer system. This is normally accomplished by obtaining information on the amount of memory the system has, both actual and virtual, along with the number of terminals hooked to the system. The kinds and types of peripheral equipment connected to the system, and the capacity of the storage devices, all provide information as to the size and capability of the computer system. This information provides a starting point from which more specific and detailed knowledge can be obtained.

The next step is to determine the computer system's multiprogramming level and obtain as much information as possible about the I/O system to include number of channels, the kinds of I/O that can be performed, along with the size of data that can be transferred. Obtain information about the speed and capacity of input and output devices such as card readers and lineprinters, along with the execution speed of the Central Processing Unit. Much of this information can be obtained from the installation manager and the rest can be obtained from the documentation. Figure 4 is a Data Flow Diagram depicting the information needed about hardware.

In gathering information about the hardware, the members of the CPE team should look for things that could have an adverse impact on performance. Such things as a slow card reader and 80% of all jobs are being entered via cards or lack of sufficient disk space which forces excess use of tape. Appendix E contains a list of things to look for. Becoming familiar with first the organization and then the workload will greatly assist in identifying problems with the hardware that could cause poor performance. Appendix F is a list of questions that will assist the members of the CPE team better understand the hardware.

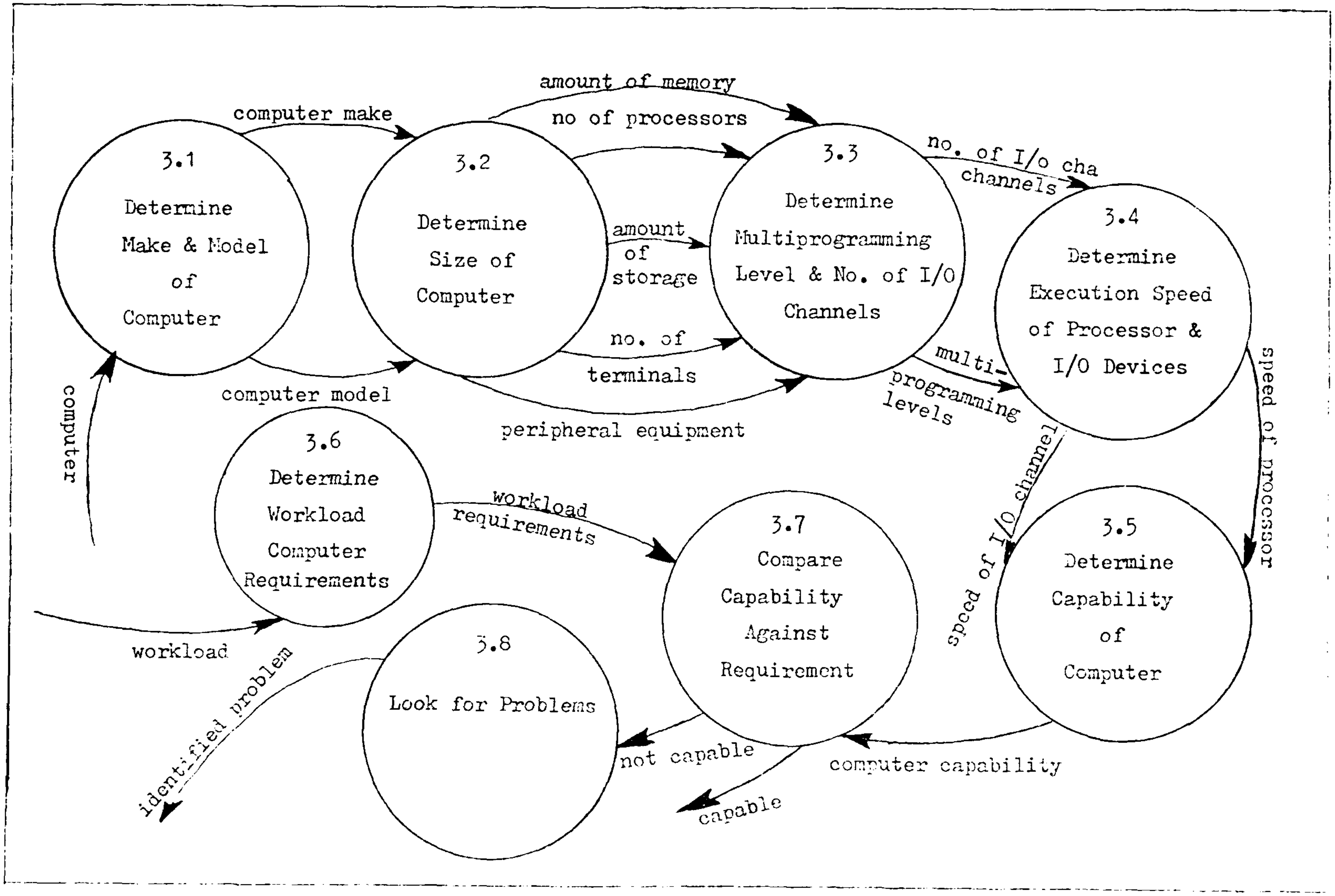


Figure 4 Data Flow Diagram to Understand the Hardware

Software

The software portion of the computer system that is referred to in this paper is the operating system. The information presented in this section should increase the CPE members' knowledge of an operating system, as well as assist them in identifying operating system problems. An operating system is a collection of programs (algorithms) designed to manage the system's resources; namely, memory, processors, devices, and information (programs and data). Some general functions of an operating system are to:

1. Keep track of the resources.
2. Enforce policy that determines who gets what, when, and how much.
3. Allocate the resource.
4. Reclaim the resource. (Ref. 8: 8)

Operating system software is the group of programs that monitor and control the operation of the computer system while the application programs are running. These monitoring and control functions include:

- Scheduling and supervising program execution.
- Allocating and releasing storage, input and output devices and other resources of the computer system.
- Controlling all input and output operations.
- Handling errors.
- Coordinating exchange of information between the computer operator and the computer system.
- Maintaining accountability of resources used by the various programs. (Ref. 20: 21)

The algorithms used by the various functions and the interactions between the different levels play an important role in the performance of the computer system. For example, an operating system that utilizes a static memory partition scheme tends to "waste" more memory than an operating system that utilizes a dynamic memory partition scheme. (Ref. 8: 116) Therefore, the more familiar a manager is with the operating system and the procedures and algorithms it employs, the better they will be able to answer questions concerning its performance. Figure 5 is a data flow diagram that can assist the members of a CPE team trying to understand an operating system.

The first step in understanding an operating system is to determine where a job goes after it enters the system. The next step is to determine how a job gets allocated and deallocated memory. Determine how the job scheduler and process scheduler works and obtain information on what queue a job can enter; also, determine how jobs leave these queues. Obtain information on how the operating system handles I/O and find out how long a job is allowed to execute. Determine the number and capability of I/O controllers or channels. While gathering this information, attempt to identify areas that could adversely impact performance, such as the majority of jobs require I/O but the system only has one I/O controller, or the system has several I/O controllers but slow peripherals such as line printers. Appendix G contains a list of questions that should assist the members identify problems

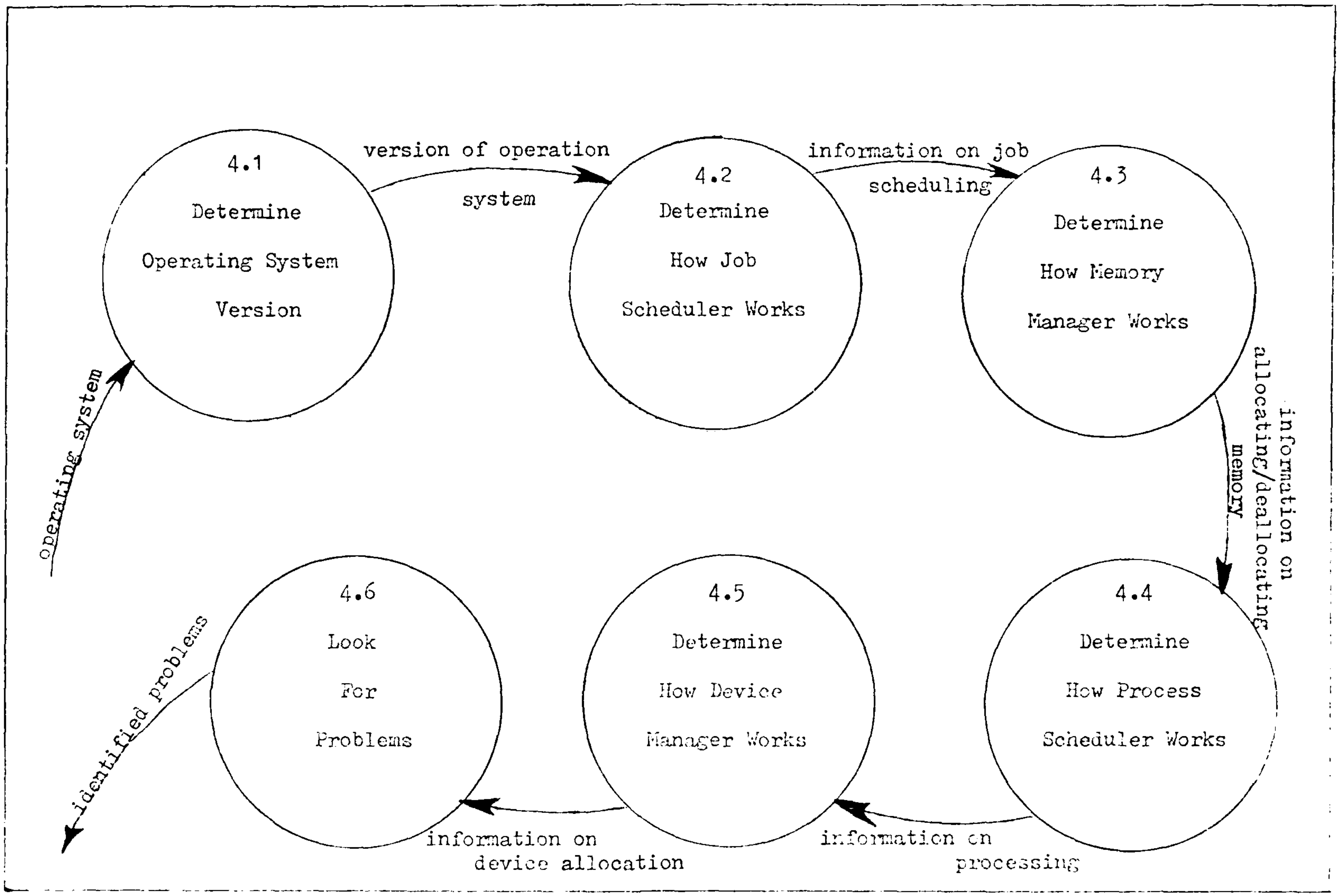


Figure 5 Data Flow Diagram to Understand the Operating System

with software. By developing an understanding of how the operating system works, the members of the CPE team become more aware of problems that could occur Once a problem occurs; however, the members must turn to other means to confirm that the problem exist. Appendix J is a list of tools and techniques that can be used to identify and confirm computer performance problems. An important point to mention here is that not all tools and techniques can be used on all systems. This is particularly true of software monitors. Therefore, before someone selects a tool to be used in a performance evaluation, they should be sure that it can be used on the computer system. (Ref. 10: VI - 50)

Interactions

So far this discussion has presented the levels of an information processing system as the organization, the workload, and the computer system. Although these levels are presented and discussed separately, their interactions play a very important role in performance analysis. Figure 6 presents these levels in diagram form.

The performance of a system is determined by the performance of individual system elements such as personnel, operating rules, amount of disk storage, etc., and the way these elements are connected into a system. Thus, any and all of the described levels contribute toward the performance seen by the user. In order to meet specific performance objectives, all of these levels have to be taken into consideration in an evaluation of performance. In

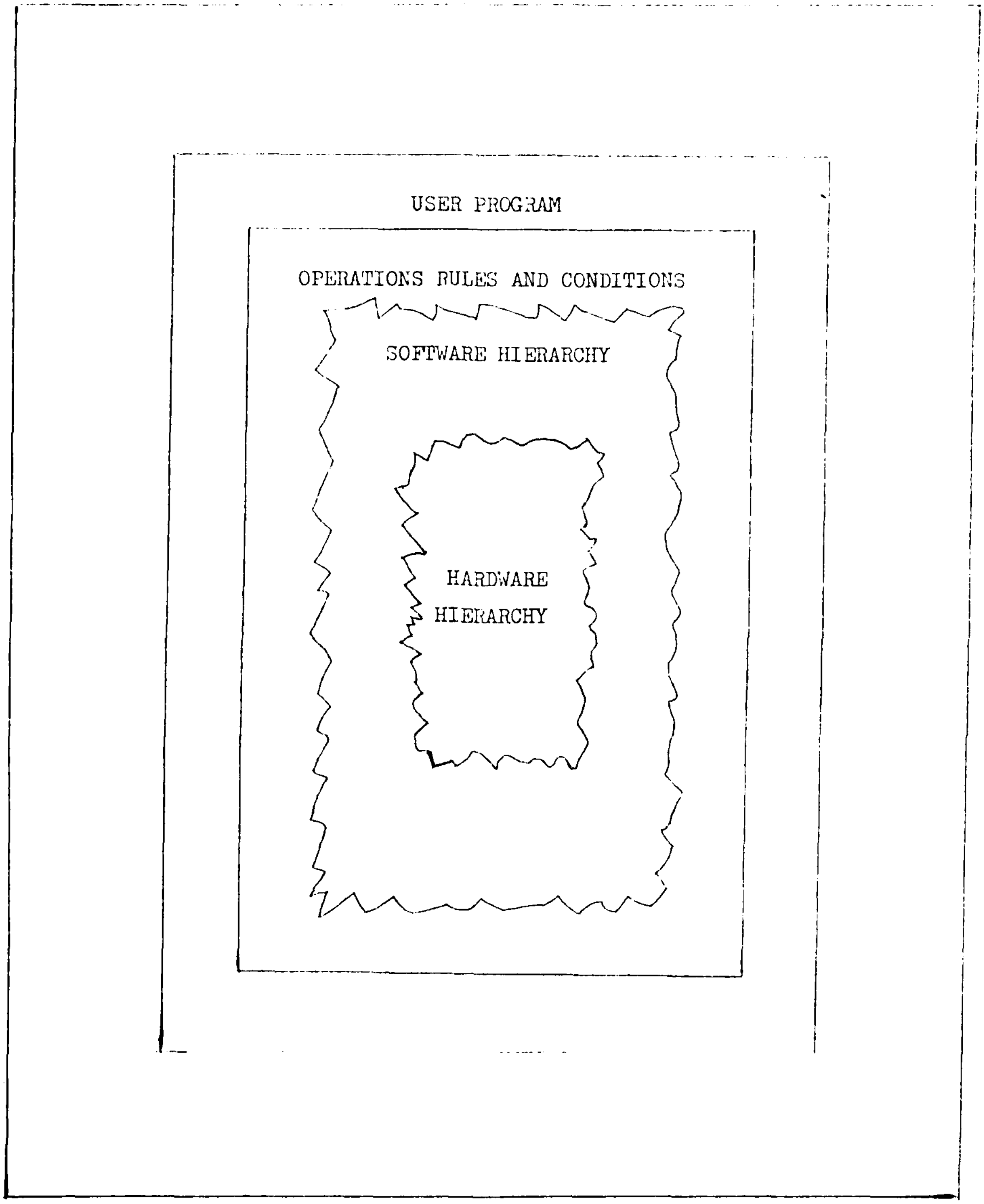


Figure 6 Levels of an Information Processing System (Ref. 20: 5)

addition, it is necessary to make a clear distinction between the system and its environment and to specify the level at which the system is to be evaluated. For this paper, the information processing system can be perceived as a total complex organization of hardware, software, users, programmers, and operators; together with the operation rules and conditions governing the interactions of the human with the non-human elements (job submitting policies, handling of tapes and disk packs requested by a job, equipment layout, etc.) (Ref. 13: 5)

Therefore, the members of the CPE team will have to become familiar with and understand the interactions of these parts of an installation in addition to understanding them separately.

Identifying problems with the organization, workload, and computer system hardware/software can be a difficult requirement for some members of the CPE team. These members may not know what to look for or where to begin, depending on their knowledge and experience. This chapter was written to provide these members with information that can help them get started and know what to look for. By using this information, the members of the CPE team should be able to broaden their understanding of the various parts of a computer center and identify problems that could cause poor performance. Additionally, this information could be used as a training tool for new members or anyone wishing a better understanding of the organization, workload, or computer system hardware/software.

III. Design of a CPE Management System

This chapter discusses the people and reports of the CPE management system. Directly stated, a computer performance evaluation management system is a structured program for continuously acquiring, analyzing, and reporting the key factors affecting the operation of the data center. (Ref. 7: VI - 201) It is concerned with:

- the identification and establishment of ADP organization and functional objectives
- the allocation of assigned resources among functional elements
- the identification of appropriate measures
- capturing data associated with performance
- analyzing performance data
- reporting performance data (Ref. 16: 6)

The main objective of the system is to present precise information about the performance of the organization, to include the computer system, on a timely basis and in a way that is understandable to management. The following is a list for developing a computer performance evaluation management system. (Ref. 7: VI - 228)

1. Recognize the Need for Performance Management.
2. Set up a Computer Performance Evaluation Team
3. Establish Performance Management Objectives
4. Define Measures and Reports
5. Administration of Group and Project Management
6. Monthly Reports, Planning, and Review

1. Recognize the Need for Performance

Before any program, system, or improvement effort can take effect, a need must arise. Unfortunately, in the field of computer performance evaluation, this need often is the result of problems with computer performance. Too often, managers of computer installations wait and delay performance management efforts until it is too late. They wait until a problem has already occurred before they start a computer performance evaluation rather than establish a computer performance evaluation management system when there are no problems and possibly identify problems before they occur.

There are many needs for computer performance evaluation. Some of these are attempting to gain additional computing time, increasing transaction volume, attempting to reduce response time, and attempting to identify performance overruns. This list is by no means exhaustive. The main point is that a need must arise and this author hopes that the need, as the case for this study, comes not from an existing problem but from management's desire to understand how their system is performing and to identify problems before they occur.

2. Set Up a Performance Management Group

The first step that management must take to implement a computer performance evaluation system is to get all people within the organization thinking about performance. This can be accomplished either at an office meeting in which all people attend or by a policy letter that will be distributed to all personnel. The main point to get across is that management is concerned with increasing the performance of the organization and the computer system; also, it encourages all people to make known problems they may have identified. These problems should be identified to the individual's immediate supervisor, who in turn will forward it to a computer performance evaluation team of specialized people. The make-up of this CPE team will largely depend upon the make-up of the organization. For example, a CPE team might be composed of an installation manager and a manager from each of the lower sections such as operations, job scheduling, or programming.

In choosing members for this team, a manager should look for several key traits. One trait is experience in systems programming and another is a scientific education. A scientific background nearly always exposes an individual to fundamentals of mathematics and statistics. These traits of systematic thinking and a knowledge of math and statistics are extremely necessary for members of a CPE group. These are not the only requirements. People with backgrounds such as applications programming, operations and operations research are also needed. The reason for this is that CPE encompasses a wide scope (design, programming, operations, engineering, analysis) and requires a mix of talents. Another trait to look for is an academically diversified background. Reasoning here is like in any other developing field the broader the members background, the more likely that parallels will be seen in other fields where similiar problems have already been solved. Since this group is responsible for evaluating the whole organization, an individual familiar with operations, as well as an individual who understands the workload, should also be included in the group. Next, and at times most important, is a person who can explain and convince. An "evangelist" would come closer to describing this person than "salesman", but evangelist would be a misleading title since no divine guidance is necessary for CPE success; however, it could help. This person is particularly important at the beginning and end of each CPE project. The best team members are open minded and eager to examine suggested changes from whatever source. Be careful that an individual who is too conservative in his work does not defeat the purpose of your CPE team. CPE is an imaginative and innovative field. It is also a field lacking in theory and strong on sharing. The "not invented here" syndrome cannot exist in a CPE team. A CPE team that does not interact with and borrow from other installations is a team that is prima facie, inefficient, and unprofessional. Choosing members for the CPE team will be a difficult task but the advantage of having a team far outweigh any difficulties that occurred during the preliminary stages. (Ref. 6: II-1, II-5)

3. Establish Performance Management Objectives

Even though the performance management effort is properly staffed and structured, it cannot proceed without knowing the direction to take; that is, the desired results must be defined. This will be dictated primarily by the charter of the data center. Typically, there is one overriding concern, such as capacity or service levels (either real-time and/or batch) or cost reductions. The actual application of the measurement team will include some combination of the above as tempered by management. The importance of management participation in defining these objectives cannot be over stressed. A measurement analyst can only apply his imagination and initiative if the results he is aiming for have been unequivocally defined. These goals, defined at the outset, should be reviewed regularly and adjusted as changes in the role of the data center occur. (Ref. 7: VI-206)

4. Define Measures and Reports

Measures and reports are the products produced by the CPE team. These products are produced in accordance with management's concerns. In determining things to measure, one should be careful of obfuscatory measurement. Obfuscatory measurement is measurement which obscures that which it should illuminate. Succinctly stated, obfuscatory measures, measure:

- the wrong things
- the right things - wrongly
- something else (i.e. other than that which they purport to measure)
- nothing at all (or at least, no meaningful thing)

Some general rules, techniques, and principles to follow when selecting things to measure are:

1. Select your measures with care - not all measures are appropriate to all situations. Tailor your measures to the tractability of your users and the gullibility of your upper management and always have a couple of reserve, just in case.
2. When in doubt, seek the advice of your mainframe vendor. Your vendor cannot sell you additional equipment until your upper management is convinced of the saturation and effective utilization of your existing configuration.
3. The easier a measure is to obtain, the more likely it is to be obfuscatory. This is one of the few cases known to modern science where Murphy's Law operates in favor of the practitioner. (There is no great mystery here: however, it is often the generosity of the vendor which made the measure easy to obtain.) Two specific kinds of measure are worthy of individual mention: means and median. Although the mean and the median each provides a single number to represent an entire set of data, the mean is usually preferred in problems of estimation and other problems of statistical inference. An intuitive reason for preferring the mean is that the median does not utilize all the information contained in the observations. A related reason is that the median is generally subject to greater chance fluctuations, that is, it is apt to vary more from sample to sample.

4. Overextend analogies - concepts which are meaningful in other fields can sometimes be transferred into the computer performance arena, where they are invalid, without loss of prestige.

5. Creative Definition - this is an indispensable element of the obfuscator's arsenal. . . for the most misleading percentage you can devise will not help you unless you can convince someone that it measures something. (Remember, "CPE efficiency"? Was there anything efficient about it?) (Ref. 12: 425-426)

The following is a list of reports and measures that can be used. Determining the reports and measures to use in the management system should be the responsibility of the installation manager and the members of the CPE team.

Utilization Report: Utilization reports provide a basis for management to see and understand equipment usage, usage of equipment dollars, and exceptional conditions in equipment usage. This information and understanding are necessities for management to assure high levels of performance. They:

- permit identification of the opportunity for performance improvement.
- provide a basis for tracking of system performance and performance improvements.
- establish confidence that cost performance is under control.
- show available operating margins and will submit the need for additional capacity.

Unused Capacity Report: Unused capacity may be described positively as "available resource" or negatively as "excess". More accurately, unused capacity is the available operating margin during the months peak period of utilization. The existance of an operating margin should not be interpreted negatively. It can clearly be a necessity. For example, a margin must be available before significant new work can be added. A large unused capacity indicates a potential cost performance improvement (i.e. the potential to perform more within current costs or reduce costs).

Estimated Limits: Computer systems cannot always be used at 100% of capacity. Often serious penalties in service and performance can result from attempts to utilize equipment at near 100% utilization. Estimated limits are best described as a performance margin. In contrast to unused capacity which is a margin of available capacity and could realistically be used; estimated limits is a capacity margin that one should not plan to use if at all possible. The word "estimated" is important. The utilization limit is not automatically derived but rather is entered manually for each component group. It may be derived rigorously through the use of measurement, simulation, or mathematics or it may be literally estimated based on observation and experience.

Peak Requirements: Peak requirements represent the amount of equipment needed as a reserve capacity to handle the actual peak load. This equipment is not used on the average but is needed for peaks. If peak requirements are small, the workload is fairly uniform from day to day. If they are large (e.g. larger than System Utilization), then the workload is variable with at least one peak period much higher than the average. The time period used to measure peak load is usually a work shift, but selection of the period is under user control.

Detection of Equipment Down Time Report: In data centers having increasingly complex equipment and configuration with multiple paths (i.e. channels and controllers) to I/O devices and often with multiple CPU's and shared device pools, detection of down time and status change is an important operations function.

Control units can fail and the software will automatically make use of alternate paths. Sometimes no operator notification is made by the software. In other cases a brief message can go undetected. Performance can degrade and thruput will be down. This situation can go undetected for hours. In addition there are a number of less critical and yet important status changes in a system that could cause problems.

- Printer out of paper
- Punch out of cards
- High speed data link active or perhaps inactive
- Disk control 1, 2, or 3 down
- Tape control 1, 2, or 3, 4 down

Trend and Variance Reports: Hardware planning reports include many variations of these basic formats: Projected Work, Projected Component Load, and Component Effective Utilization Threshold.

Depicting level of activity as a function of time, they allow broadbased planning functions to make predictions. The applied value of the performance management system is in charting actual versus projected activity, enabling planners to refine their estimates by applying known performance data to calculated performance levels. In this way, Margin Analysis Reports will aid planners and provide additional information for management by control limits. (Ref. 7: VI 236-253) Figure 7 shows an example of a margin analysis report.

Measures

Measures used in the computer performance evaluation system are of two types; measures of effectiveness and measures of efficiency. Measures of effectiveness define how well the organization's objectives are being accomplished while measures of efficiency define how well the organization is utilizing assigned resources. Measures of effectiveness are:

- Timeliness: The extent to which the objective is accomplished in time to be effective. Typical measures include schedule performance, turnaround, and response time.
- Quality: The extent to which the objectives are accomplished acceptably. Typical measures include accuracy, usefulness, clarity, and acceptability.
- Quantity: The extent to which the objective is satisfied. Typical measures include volume achieved and amount of backlog.

Measures of efficiency are categorized as follow:

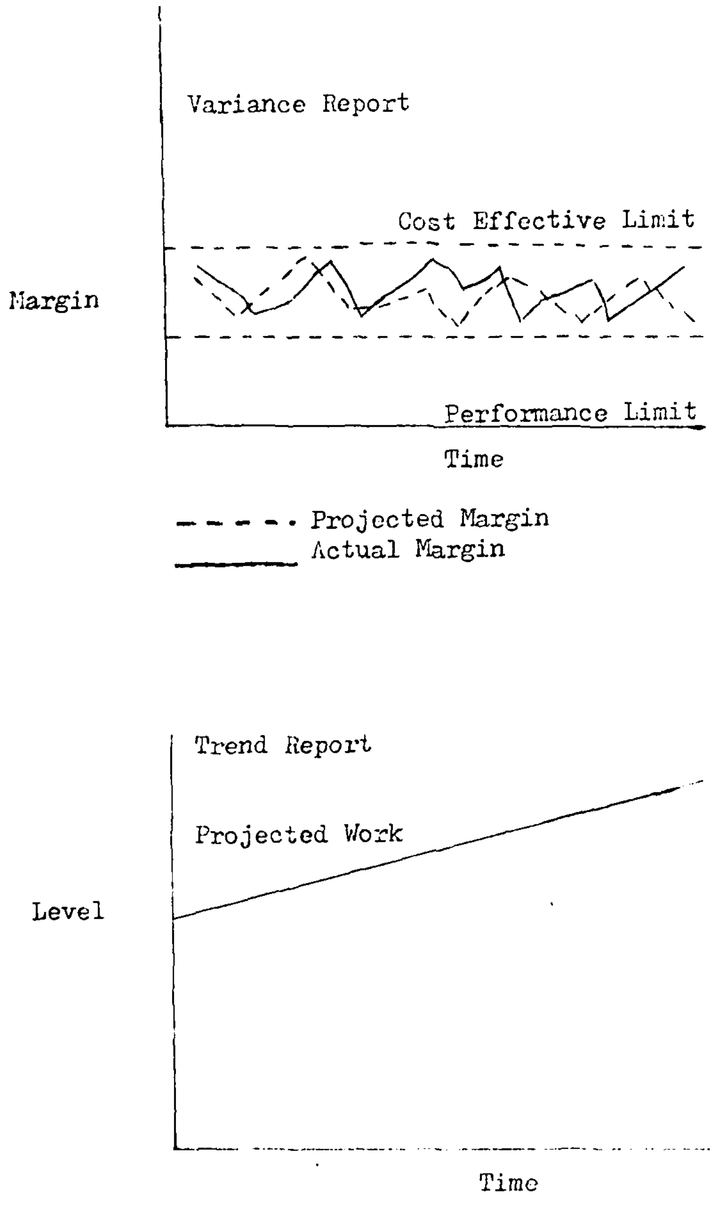


Figure 7 Margin Review and Analysis Report (Ref: 7: VI 252, 253)

- Staff: The amount of human resources utilized per unit of output. These resources can be further classified by skills level, skills mix, and function performed.
- Machine: The amount of equipment employed. These can be classified by size, capacity, type, and arrangement.
- Material: Supplies consumed in the accomplishment of the objective.
- Money: All resources can be converted to a single common denominator, money. Conversely, the availability of this resource can be converted into required resources.

While other classification systems are possible, this classification of effectiveness and efficiency measures proves to be a useful structure for the systematic identification of a computer performance management system.

5. Administration of Group and Project Management

Depending on the size and complexity of the computer system and the organization, some managers may want to create additional positions and hire additional people to make up the CPE team. Job descriptions for these individuals should mention:

- Work with documentation describing the logical and physical flow of signals through the system and its components.
- Perform measurement analysis and coding changes of applications and control programs.
- Install or connect such CPE devices or tools as are available.

- Develop and use simulation and modeling techniques.
- Document activities and recommendations for use by management.

These may be elaborate as required to fit each personnel department's peculiar needs, but these five basic areas are necessary to insure that the CPE team has the minimal capability to performm useful projects. (Ref. 6: II-3)

The day-to-day operating procedures of the CPE team are essentially the same as for an audit group: define a problem, examine existing methods, postulate changes, test change, recommend best changes, and oversee implementation of accepted recommendations. Once a problem is identified, the CPE team can turn it into a project. Since some team members have specific skills that not all team members have, each project or study may have a different project manager. A project manager's job in classical management terms to:

- Plan: Determine requirements of the project in terms of people, facilities, equipment, etc., estimate the project's duration, and relationships between workload of the CPE team and the project's deliverables.
- Organize: Request and insure availability of the necessary facilities, equipment, personnel, material, and (most important) authority.
- Direct: Set time and cost milestones and bounds for what must be done; make all operating, project-specified decisions.

- Control: Measure performance against plan and take away any necessary actions to correct malperformance.
- Coordinate: Insure that all involved activities are aware of and receptive to the projects' efforts, goals, and recommendations.

Appendix I is a detailed list of a project manager's responsibilities. The last duty of the project manager is to present his findings and recommendations to his boss. This leads us to a very important question; to whom does the computer performance evaluation team report. (Ref. 12: II-9)

Because organizations are so very different, the only way of describing placement in general, is in relative terms. The team should, at the very lowest, report to a person that has the authority to implement the teams proposed recommendations. This placement insures that recommendations that are accepted will be implemented. Since the computer performance evaluation team will be dealing with problems of a wide spectrum, they will need information on all scheduled and unscheduled changes. To insure they receive this information, a charter should be established.

The charter will allow the team to examine both existing systems and proposed changes before new workloads or equipment acquisitions are committed. Determining the impact of such changes requires a thorough knowledge of existing workloads and systems. The benefit from this initial learning phase for the CPE team will be significant. It is always the first time that a review of all facets of the computer center's activity will have been attempted by knowleegeable and impartial persons. A charter should include, as a minimum, the following statements.

> "No new programs or systems of programs and no substantial changes to existing programs are to be implemented without first having the design or change reviewed by the CPE team."
>
> "Before ordering new equipment to replace, to add to, or to enhance existing equipment's speed or capacity, the CPE team must be called upon to measure the levels of activity and contentions of the system or portions of the system that would be affected by the new equipment." (Ref. 6: II-6, II-15)

Once established, the CPE team loops through a series of continuing activities.

A. Examine and select new CPE tools to keep up with developments.

B. Interchange information with other CPE teams, attend professional meetings, and read the technical literature.

C. Determine CPE measurement frequencies with respect to:

1. The characteristics of each installation
   a. Workload stability
   b. Personnel and system changes
2. The aging systems
3. Planned workload changes

a. New systems

b. New users

D. Integrate operations and measurement activities; develop a "performance consciousness" at all levels.

1. Use existing and new operations reports

   a. Console logs

   b. System Incidence Reports

   c. Suggestion Box

2. Educate operations personnel to be aware of visual cues to system malperformance.

3. Educate operators in use of tools to improve selection of jobs for multiprogramming, mounting of tapes, and disk packs upon request, etc.

E. Establish good relations with the computer system vendor.

1. You may require his assistance for probe point development.

2. You may require his permission to attach a hardware monitor.

3. First instinct of maintenance personnel is to resist measurement activities.

4. Acquire a knowledge of his activities and the acquaintance of his research staff.

The computer performance evaluation management system should be viewed from a total systems perspective. It should be used in a life cycle context, from procurement through operations, from design evaluation through operational tuning, from procedures evaluation to the establishment of job control language standards.

CPE should be used to achieve the level of service required by management at the lowest total system cost - - it should not be concerned with techniques but with performance.

6. Monthly Reports, Planning, and Review

Once the system is established and working, it will be of little value unless management can see some results or reports. These reports will be the only real (tangible) product of a properly run CPE team. Some of the information contained in the reports needs to be collected over a certain period of time such as a month or week before the information will be of any value to management. For example, to say that the computer processed for twelve hours on Tuesday has little meaning when compared to 180 hours of computer processing for the month of June. An example of some monthly reports are utilization reports, trend reports, and margin and variance reports. An example of some weekly reports are system profiles, CPU utilization by shift, average channel utilization by shift and peak loading. (Ref. 7: VI-240) The actual periods to be covered by the reports should be determined by the specific needs of each organization or installation. Some reports should be scheduled to cover relatively long periods (annual, semi-annual); others should cover relatively short periods (weekly, monthly).

Generally, management has three requirements for performance reporting:

- One time report: First time that performance management is introduced to the organization or for special situations.

- Long time Periodic reports: To present performance where few events occur over long periods.
- Short term periodic reports: To present performance where many events occur over short periods.

These three requirements can be used to define the reports that should be produced by the computer performance evaluation system. The information contained in these reports will be used by managers to answer questions concerning the organization's growth and performance. (Ref. 16 : 174)

Since an organization's computer system and mission can change, procedures need to be established by management where they review the pertinency of the reports they are receiving. For example, reports showing an I/O controller busy 90% of the time would not be valid if the organization had just connected a second I/O controller. Ideally, the reports should be reviewed everytime a change occurs in the workload and computer system. This insures that management is reviewing the latest and most accurate performance information on their computer system.

Tradeoffs

The computer performance management system which has just been presented may not be appropriate for all organizations. The major drawback is that some organizations may not have the personnel needed to establish a CPE team. Another drawback is that some managers may not see the need for computer performance management. A manager of an organization that does not see the need for computer performance management is either new to the field or has worked for an organization with countless funds (i.e. whenever hardware/software is needed they purchase it), or one that has experienced little or no growth (i.e. their computer system has always been able to handle the workload). Unfortunately, little can be done for the manager who does not recognize the need for performance management. There is however, help for managers that are interested in CPE and know how important it is but, because of work requirements cannot assign CPE responsibilities to individuals within their organizations. The help for these managers lies in the types of analysis they can implement. These managers can implement a partial analysis rather than a full analysis. A full analysis requires three to four members of a CPE team to look at all aspects of the computer system and its environment. A partial analysis requires only one member to be on the CPE team and this person focuses on a certain aspect of the computer system rather than the entire computer system and its environment. The advantage of a partial analysis is that it requires less time, effort, and people than a full analysis. The major disadvantage is that the responsiblity of the analysis falls upon one person and that person is normally the system manager. Another disadvantage is that while this person is focusing on one aspect of the computer system and trying to solve a particular problem, other problems may be occuring which could very well offset the actions taken to correct the initial problem. In addition, this person will probably be forced to work with the existing information being gathered on computer performance rather than obtaining other tools to gather different kinds of information which could identify other problems. Even though the disadvantages far outweigh the advantages, many installation and system managers will be forced, because of work requirements, to implement a partial, rather than a full analysis. Therefore, to insure that a system or installation manager receives the most from the partial analysis, some general guidelines should be followed

The first guideline to be followed is to review information that requires little time and effort but contains lots of computer performance data. Information from an accounting package is a good example. The frequency with which this information is reviewed will largely depend on the organization and the person performing the analysis. Next, the person should determine if there is any other information different from that presented by the accounting package that can be gathered from or by the computer system that could show additional computer performance data. An example of this would be something similiar to the display utility on Digital Equipment Corporation's VAX II/780 which provides CPU and I/O information. Although the information and reports that a partial analysis provides are limited, they may be all that some organization need. These organizations could be classified as those with limited growth potential or those with unlimited funds where the computer system is always larger than really needed.

On the other side of the coin though, are the organizations without countless funds and ones that are experiencing normal or above normal growth. For these organizations, a partial analysis would probably not meet the requirements for computer performance measurement. This would be even more true if the organization was already or had been experiencing computer performance problems. To lay the burdens of identifying problems, solving existing problems, and determining future requirements and constraints on one individual is a little unreasonable. These organizations could definitely benefit from implementing a full analysis. Although the cost of a full analysis in terms of people, time, and resources and possibly money is considerably more than a partial analysis, the benefits it can provide should prove to be cost effective in the long run.

This chapter presented guidelines to follow and methods to use when establishing a CPE team and when defining reports. Since the CPE team is the most essential part of the CPE management system, defining their qualifications cannot be overstressed. The better educated and experienced the members of the CPE are, the more capable they will be to identify and solve computer performance problems.

IV. APPLICATION

After a CPE team has been established, the implementation of the CPE management system becomes an easier task. By using the guidance, figures, and questionaires of Chapter 2, the CPE team members can immediately start obtaining information about the computer installation and start looking for performance problems. This chapter shows detailed information that can be obtained about a computer installation by using the procedures presented in Chapter 2.

SEAFAC Organization

The Systems Engineering Avionics Facility (SEAFAC) is a division of the Aeronautical System's Division. SEAFAC is presently in the process of developing and constructing a hot bench to simulate the flight of the KC-135 aircraft. The hot bench consist of a mission computer, the VAX 11/780, and software which simulates sensor inputs and other aspects of the flight environment. The VAX 11/780 digital computer accepts hot bench software inputs from a MIL-STD-1553 Data Bus and the operator's console. Stored in the computer's memory are five simulation programs: master executive, simulation executive, control head, monitor, and display. All of these programs compete for a single processor along with other programs such as the terminal driver. The VAX 11/780 computer is responsible for overall hot bench control, control of data display and a means for running the real time simulation. (Ref. 5: 160)

In a typical "flight" of the hot bench, the flight computer accomplishes such functions as cockpit management, flight planning, navigation functions, and fuel management. Software simulation models, which are in fact Fortran programs, provide sensor information to the flight computer and provide the environment for the flight computer to "fly in". Table 1 briefly outlines the signals as they apply to the simulation.

The reasons which provoked this study were management's interest in measuring the performance of the VAX 11/780 during software development and running the real-time simulations. An organization chart of SEAFAC is presented in Figure 8.

The computer system used by SEAFAC is operational twenty-four hours a day; however, all software development and simulations take place between the hours of 0800 and 1700 Monday through Friday. The organization presently has a mix of government and contract programmers. These individuals are developing software for the KC 135 simulation. All software development is accomplished on line via VT-100 terminals; as such the organization has only one operator. This individual is responsible for loading paper in the printers and informing management of problems with the system. No shifts are required and the organization does not have a resident maintenance technician. The organization does however, have a resident Digital Equipment Corporation systems programmer who provides assistance in highly technical matters.

The management levels within the organization are determined by the project. For example, on the KC-135 project, individuals were assigned management positions based on knowledge and

TABLE 1

Input/Output Signals to VAX 11/780

For KC-135 Hot Bench

Signal	Sim. Program Called	Function
Performance Monitor	Simulation Executive	Call the Simulation Executive which in turn calls the models to be run to simulate the hot bench environment.
Operator Commands	Master Executive	Commands issued by the hot bench operator which actually control all elements of the hot bench. These commands include: Start, Stop, Freeze, etc.
Fault Signals	Master Executive	These commands allow the operator to deliberately introcuce fault conditions into the system.
Control Head Signals	Control Head	Establish system controls such as those normally input by the pilot via the yoke.
Display Signals	Display	Provide inputs to the display simulation which controls numeric displays for the hot bench.
Altimeter Signals	Display	Flight level inputs.
Clock Signals	Display	Shows simulation time to the tenth of a second.
Monitor Signals	Monitor	Make Bus data available to display elements.

(Ref. 5: 159)

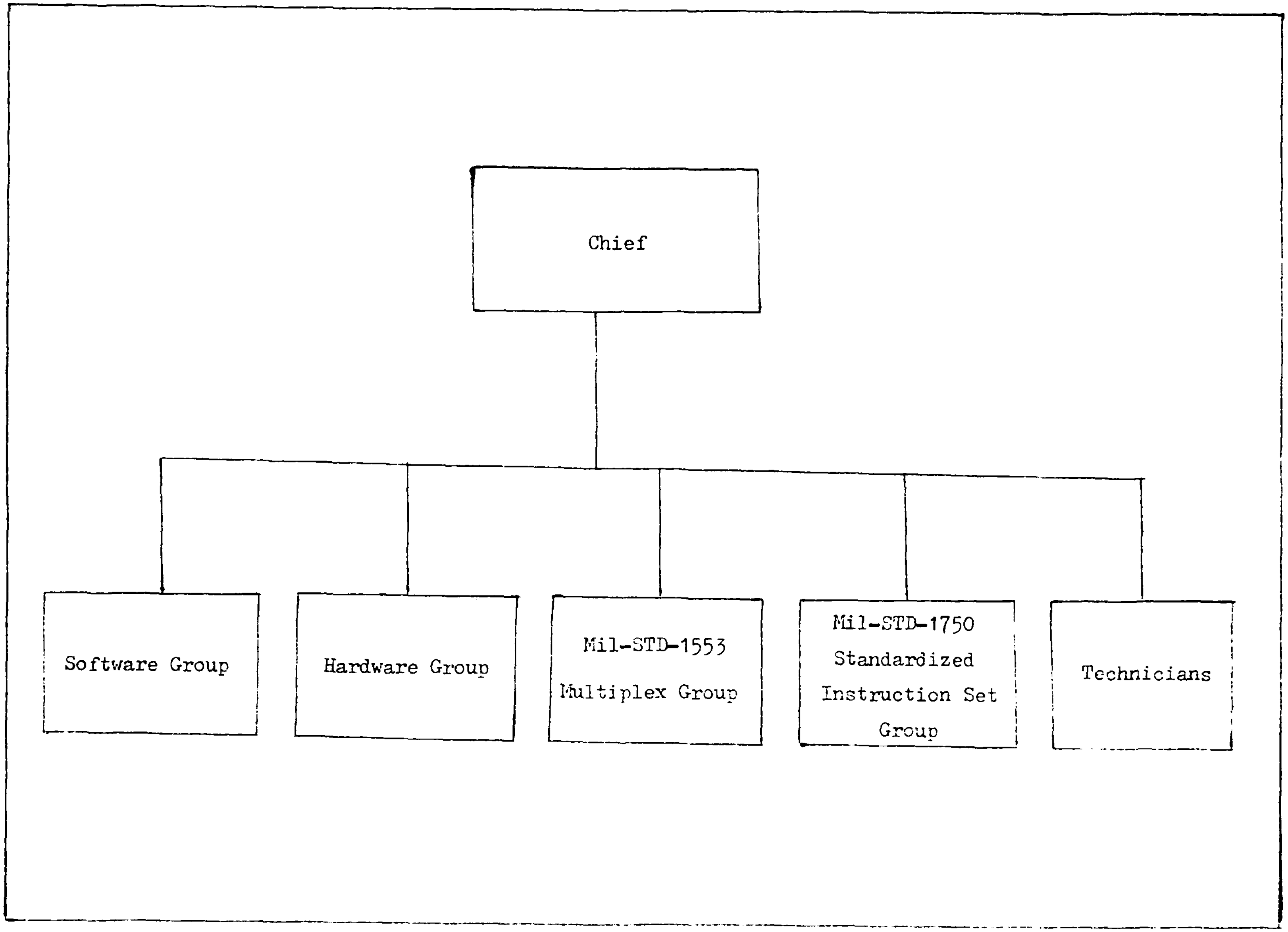


Figure 8 SEAFAC Organization Chart

experience in such areas as hardware, software, MIL-STD-1553 and MIL-STD-1750. Individuals in charge of these areas work together and within their group to insure all aspects of the project are completed on or near the scheduled due date.

Managers within the organization have established their own procedures whereby they receive progress reports on the status of particular projects. Little performance information is gathered on or about the VAX 11/780 and managers have minimal knowledge of its performance. This, coupled with the fact that the programmers consider the computer a free good, were the only problems observed in the organization.

SEAFAC Workload

SEAFAC is responsible for basically two types of workload; software development and real time simulations. In the software development phase, the programmers develop and test software that will be used during the simulation phase. This software will be classified by the module the programmer is trying to build. Each module will perform a different function during the simulation. After a programmer has completed and successfully tested a portion of a module or specific program, he presents it to the manager of the group, who runs the program against a program analyzer to determine its efficiency. During software development, all programmers have the same priority and compete for computer resources.

The system jobs run by the VAX are memory management, process management, device management, and input/output services. Since the memory and disk available on this machine are sufficiently large the above areas pose no problems in terms of load. Also, since the organization has, at present, twenty five programmers and not all of them program at once, the load caused by these programmers is insignificant. Because the programmers are developing software on a fairly large machine, i.e. four meg actual memory and four RM-03 disks. job scheduling and processing time do not cause problems as in some organizations. Additionally, backlogs never occur and the only problems observed with software development were programmers utilizing too much disk space and occasionally some projects were completed late. These late projects were caused, not by problems with the hardware/software, but seemed to be caused by lack of sufficient personnel. Other problems may be identified during the actual running of the simulation; however, at this time the organization is still in the software development phase of this project.

During a simulation, the VAX computer will be used to simulate an aircraft. Housed in the VAX's memory will be certain modules that perform such functions as cockpit management, flight planning, fuel management, and navigation functions. The programs that compose these modules will have a different priority and will compete for the single processor. During an actual simulation job scheduling, job processing, memory management, and I/O services will all play a very important part in the performance of the simulation. As such managers need a general understanding of how each of these functions perform their specific tasks. More will be presented on this in the software section of this chapter.

The main problem observed while studying the workload was that little information was being gathered about the workload by the system. The reason for this was that the system was basically a "free good". Another reason was the lack of a sufficient accounting package. Although the organization has an accounting package, it is extremely limited. Table II presents the parameters that the accounting package gathers data on. The accounting package was written by a Digital Equipment Corp User's Group (DECUS), rather than Digital Equipment Corporation and as such, documentation was extremely limited. Additionally, the accounting package was not being used. To study the impact of the workload on the computer system, the system manager uses several sources provided and maintained by the computer system. These sources will be presented and discussed later in this chapter.

SEAFAC Computer Hardware

The computer system used by SEAFAC is Digital Equipment Corporation's VAX 11/780. The VAX 11/780 is a high performance multiprogramming computer system. The system combines a 32-bit architecture, efficient memory management, and a virtual memory operating system to provide essentially unlimited program address space. The size of the VAX computer located at SEAFAC is relatively large. Their computer has four Meg of memory. This memory consists of arrays of MOS RAM integrated circuits with a cycle time of 600 nanoseconds. Disk storage for the organization is provided by four RM03 disk drives. The RM03 is a high speed, medium capacity disk drive. Its peak capacity is 67 mb with a

TABLE II

Information Gathered by Accounting System

Listing of File Account Data	Totals for File: Account Data
User Name	User Name
Type	Login - Count
Start Time	Connect - Time
Elapsed Seconds	CPE - Seconds
Page Faults	Buf I/O
Buf I/O	DIR I/O
DIR I/O	Vol - Mounts
Vol Mounts	Print - Jobs
Job - Name	Print pages
Page Count	Total Logins
Q. I/O's	Total Connect Time:
	Total CPU Seconds:
	Total Print Jobs:
	Total Pages Printed:
	Total Records Read:
	Login Failure Count:

peak transfer rate of 1200 kb/sec. The average seek time is 30 ms and average rotational latency is 8.3 ms. The system supports multiprogramming applications that require high performance by providing:

- event driven priority scheduling
- rapid process context switching
- minimum system service call overhead
- process access mode memory protection
- memory management control

The system schedules processes for execution based on the occurrence of events such as I/O completion rather than time quantum expiration. In addition, the computer system has 25 VT-100 terminals connected to it.

The VAX 11/780 central processing unit performs the logic and arithmetic operations requested by the computer system user. The processor is a high-performance, microprogrammed computer that executes a large set of variable length instructions in native mode, and nonpriviledged PDP-11 instruction in capatability mode. The CPU uses 32-bit virtual addresses, allowing access to over four gigabytes (4Gb, 2^{32}) of virtual address space. These addresses are called virtual because each address is not necessarily the actual address in physical memory. The processor's memory management hardware translates virtual addresses to physical addresses.

The processor provides sixteen 32-bit registers that can be used for temporary storage, as accumulators, index registers, and base registers. Four of these registers have special significance. They are the Program Counter and three registers that are used to provide an extensive CALL facility.

The native instruction set is highly versatile and bit-efficient. It includes integer, packed decimal, character string, bit field, and floating point instructions, as well as program control and special instructions. The VAX 11/780 processor includes an 8K byte cache, integral memory management, 32 interrupt priority levels, an intelligent console, a programmable real-time clock, and a time-of-day and date-clock. (Ref. 21: 4-1)

The peripheral systems supported by the VAX 11/780 are of four types.

- Mass storage peripherals such as disk and magnetic tapes
- Unit record peripherals such as line printers and card readers
- Terminals and terminal line interfaces
- Interprocessor communication links

All peripheral device control/status registers (CSR's) are assigned addresses in physical I/O space. No special processor instructions are needed for I/O control. In addition, all device interrupt lines are associated with locations that identify each device's interrupt service routine. When the processor is interrupted on function request completion, it immediately starts executing the appropriate interrupt service routine. There is no need to poll devices to determine which device needs service.

Devices use either one of two types of data transfer techniques; direct memory access or programmed interrupt request. The mass storage disk and magnetic tape devices and the interprocessor communications line are capable of direct memory access (DMA) data transfers. The DMA devices are also called non-processor request (NPR) devices because they can transfer large blocks of data to or from memory without processor intervention until the entire block is transferred.

The unit record peripherals and terminal interfaces are called program interrupt request devices. These devices transfer one or two bytes at a time to or from assigned locations in physical address space. Software then transfers the data to or from a buffer in physical memory. (Ref. 21: 5-1)

The VAX computer presently has two channels to disk and all I/O requests are generated by a Queue I/O (Q10) Request system service. If a program calls the record management system (RMS) procedures, RMS calls the Q10 system service on the program's behalf. Que I/O Request processing is extremely rapid because the system can:

- optimize device unit use by minimizing the code that must be executed to initiate requests and post request completion.
- optimize disk controller use by overlapping seeks with the I/O transfers.

The processor's many interrupt priority levels, increase interrupt response because they enable the software to have the minimum amount of code executing at high levels by using low priority levels for code handling request verification and completion notification. (Ref. 22: 10) Input/Output operations under VAX/VMS are designed to be as device and function-independent as possible. User processes issue I/O requests to software channels which form paths of communication with a particular device. Each process can establish its own correspondence between physical devices and channels. I/O requests are queued when they are issued and processed according to the relative priority of the process that issued them. I/O requests can be handled indirectly by the VAX Record Management Service (RMS) or they can interface directly to the VAX/VMS I/O system. For more information on VAX RMS and VAX I/O, refer to Volume 7 and Volume 4 of the DEC reference manuals.

In addition, device drivers take advantage of the processor's ability to overlap execution with I/O by enabling processes to execute between the initiation of a request and its completion. User processes can queue requests to a driver at any time and the driver immediately initiates the next request in the queue upon receiving an I/O completion interrupt.

A VAX/VMS driver performs the following functions:

- Defines the peripheral device for the rest of the VAX/VMS operating system.
- Defines the driver for the operating system

procedure that maps and loads the driver and its device data base into system virtual memory.

- Initializes the device (and/or its controller) at startup time and after a power failure.
- Translates software requests for I/O operations into device-specific commands.
- Activates the device.
- Responds to hardware interrupts generated by the device.
- Reports device errors.
- Returns data and status from the device to software.

Device drivers work in conjunction with the VAX/VMS operating system. The operating system performs all I/O processing that is unaffected by the particular specifications of the target device (i.e. device independent processing). When details of an I/O operation need to be translated into terms recognizable by a specific type of device, the operating system transfers control to a device driver (i.e. device dependent processing). Since different peripheral devices expect different commands and setups, each type of device on a VAX/VMS requires its own supporting driver. The VAX/VMS operating system contains device drivers for a number of standard DIGITAL-supported devices. These include both Massbus and Unibus devices. In addition, the user can write additional drivers for non-standard Unibus devices.
(Ref. 21: 6-18)

In comparing the workload requirements against the computer capability, it was obvious that there were no problems. This was indeed fortunate because the kinds and types of information needed to make this comparison was not and could not be gathered by the organization.

While looking for other problems, several were identified. The first problem is that the organization presently has four RM03 disks and two I/O controllers/channels and no way of measuring traffic to and from the disk; therefore, balancing the disk and making decisions such as whether to put all system files on one pack or two will be extremely difficult, if not impossible. Another problem which relates to the comparison problem presented above is the organization's inability to gather and present long term measurement information. An example of this being, hours of processing time for a particular month.

SEAFAC Computer Software

This section discusses the operating system of the VAX 11/780. The VAX 11/780 has a virtual memory operating system. VAX/VMS, as it is called, is a multiuser, multifunction virtual memory operating system that supports multiple languages, an easy to use interactive interface, and program development tool. The VAX/VMS operating system is designed for many applications, including scientific/real-time, computational, data processing, transaction processing, and batch. The operating system performs process-oriented paging which allows execution of programs that may be larger than the physical memory allocated to them. Paging is automatically handled by the system, freeing the user from any need to structure the program. In the VAX/VMS operating system, a process pages only against itself - thus individual processes cannot significantly degrade the performance of other processes. VAX/VMS schedules CPU time and memory residency on a preemptive priority basis. Thus, real-time processes do not have to compete with lower priority processes for scheduling services. Scheduling rotates among process of the same priority. The VAX/VMS operating system provides a file and record management facility that allows the user to create, access, and maintain data files and records within the files with full protection. This is accomplished by a user authorization file. A user authorization file entry describes the characteristics of a process that is associated with the user for whom the process is created. Additional information in a user authorization file entry includes:

- User name, password, and account name.
- Priviledged system functions to which the user is allowed access.
- Limits and quotas for the system resources that can be used.
- Priority of the process created for the user at login.
- Default command interpreter.
- Default disk device and directions. (Ref. 20: 4-1)

The system manager maintains a user authorization file that contains an entry for each user allowed access to the system. For more information on UAF refer to Chapter 2 of the System Manager Guide.

In addition, VAX/VMS provides a program development capability that includes editors, language processors, symbolic debugger, and on-line error logging of CPU errors, memory errors, peripheral errors, and software failures. The virtual memory operating system enables the programmer to write large programs that can execute in both small and large memory configurations without requiring the programmer to define overlaps or later modify the program to take advantage of additional memory. These are some of the general capabilities of VAX/VMS version 2.1. (Ref. 21: 2-1, 2-2)

The VAX does not have a "job scheduler" per se; rather, jobs are broken into processes. A process is the schedulable entity capable of performing computations in parallel with other processes. It consists of an address space and an execution state that define the context in which a program image executes. An executing program is associated with at least one process, but it can be associated with several processes. Each process has a base priority assigned to it when it is created. The priority of a real-time process remains unaltered by the system during the process's execution; however, a normal process is subject to having the scheduler alter its priority during the course of its execution.

The scheduler uses a modified pre-emptive priority algorithm for normal process's recent execution history. Each process has a current priority in addition to its base priority. The scheduler dynamically changes the current priority as the process executes; however, the current priority is never less than the base priority. Scheduling according to strict priority for real-time processes and using a modified priority for other processes allows the scheduler to achieve maximum overlap of computer and I/O activities while still remaining responsive to high-priority real-time requests. The scheduler uses priority increments to modify the priority of a normal process. Each system event has an assigned priority increment that is a characteristic of the source of the event. If the event causes a state change to an executable state for the process, the scheduler adds the increment to the base priority. The only restriction is that a process's current priority never decreased to a value below its base priority or increased above a priority of fifteen, and a real-time process' priority is never modified. (Ref. 22: 373-375)

The VAX/VMS scheduler performs normal and real-time process scheduling based upon the priority of the executable process in the balance set. A normal process is also referred to as a time shared or background process while a real-time process is referred to as time-critical.

VAX/VMS defines thirty two distinct levels of software priority for the purpose of scheduling. Priorities range numerically from 0-31, where 31 represents the highest software priority. The operating system allocates priorities 0-15 to the scheduling of normal processes while priorities 16-31 are dedicated to the scheduling of time-critical processes. Time-critical processes are scheduled strictly by priority; when a higher priority process is ready to execute, it preempts the process currently running. Normal processes, on the other hand, are scheduled using a modified preemtive algorithm to achieve maximum overlap of computation and I/O activities.

Time critical processes take precedence over background processes in the queue for execution since they are of higher priority. The VAX/VMS scheduler performs process scheduling functions based upon the following variables:

1- The process priority.

2- The occurrence of system events and resulting process state transitions.

3- The expiration of in-memory time allowed to a non-time critical process, i.e. quantum over flow.

System events are occurrences that cause the status of one or more processes in the system to change. The scheduler reflects the change by removing the process control block from one state queue and queuing it in the current state queue. An execution process can cause a system event by putting itself in a wait state, or it can cause a system event for another process. In addition, system components like the swapper and the timer can cause system events.

Regardless of the source, all system events are reported to the scheduler. (Ref. 22: 143-144)

The memory management technique utilized by the VAX operating system is known as virtual memory. Virtual memory refers to the concept that a program's location in main memory is transparent to the process. Additional features of the VAX-11 virtual memory scheme are:

1- Only a portion of the program (those pages which are being actively referenced) need reside in main memory during execution.

2- Programs (processes) are allowed to exceed the maximum amount of main memory available.

The memory management scheme maintains a data base called page tables describing the status of all physical pages of memory and the status and location of all virtual pages of all processes in the system. The function of memory management is to map virtual pages into physical address space, to control the paging of the working set of pages in active use by the process, and to provide per-process and inter-process memory protection. To help provide required mapping and protection; virtual address space is divided into 512 byte sections called pages. The page is the basic unit of relocation and protection. Memory management utilizes page tables as the data base to contain the status and location of virtual pages of processes. Each individual page of a process has associated with it, an entry in an appropriate page table to describe that page. (Ref. 22: 123)

The VAX/VMS device driver is a set of tables and routines that control I/O operations on a peripheral device attached to a VAX-11 system. A driver performs the following functions:

- Defines the peripheral device for the rest of the VAX/VMS operating system.
- Defines the driver for the operating system procedure that maps and loads the driver and its device data base into system virtual memory.
- Initializes the device (and/or its controller) at system startup time and after a power failure.
- Translates software requests for I/O operations into device-specific commands.
- Activates the device.
- Responds to hardware interrupts generated by the device.
- Reports device errors.
- Return data and status from the device to user software.

Device drivers work in conjunction with the VAX/VMS operating system. The operating system performs all I/O processing that is unaffected by the particular specifications of the target device (i.e. device-independent) processing. When details of an I/O operation need to be translated into terms recognizable by a specific type of device, the operating system transfers control to a device driver (i.e. device dependent processing). Since different peripheral devices expect different commands and setups, each type of device on a VAX/VMS requires its own supporting driver. A device driver contains a set of subroutines that the operating system calls to perform device-dependent processing on an I/O request. The subroutines of a VAX/VMS driver perform the following functions:

Initialization: At the time that the driver is loaded or after a power failure, initialize a device or controller by setting hardware registers and initializing fields in the I/O data base.

I/O Setup: Prepare an I/O request for a device for formatting data, allocating system buffers, locking pages in memory, etc.

I/O Startup: Set up device registers and the I/O data base to start a device.

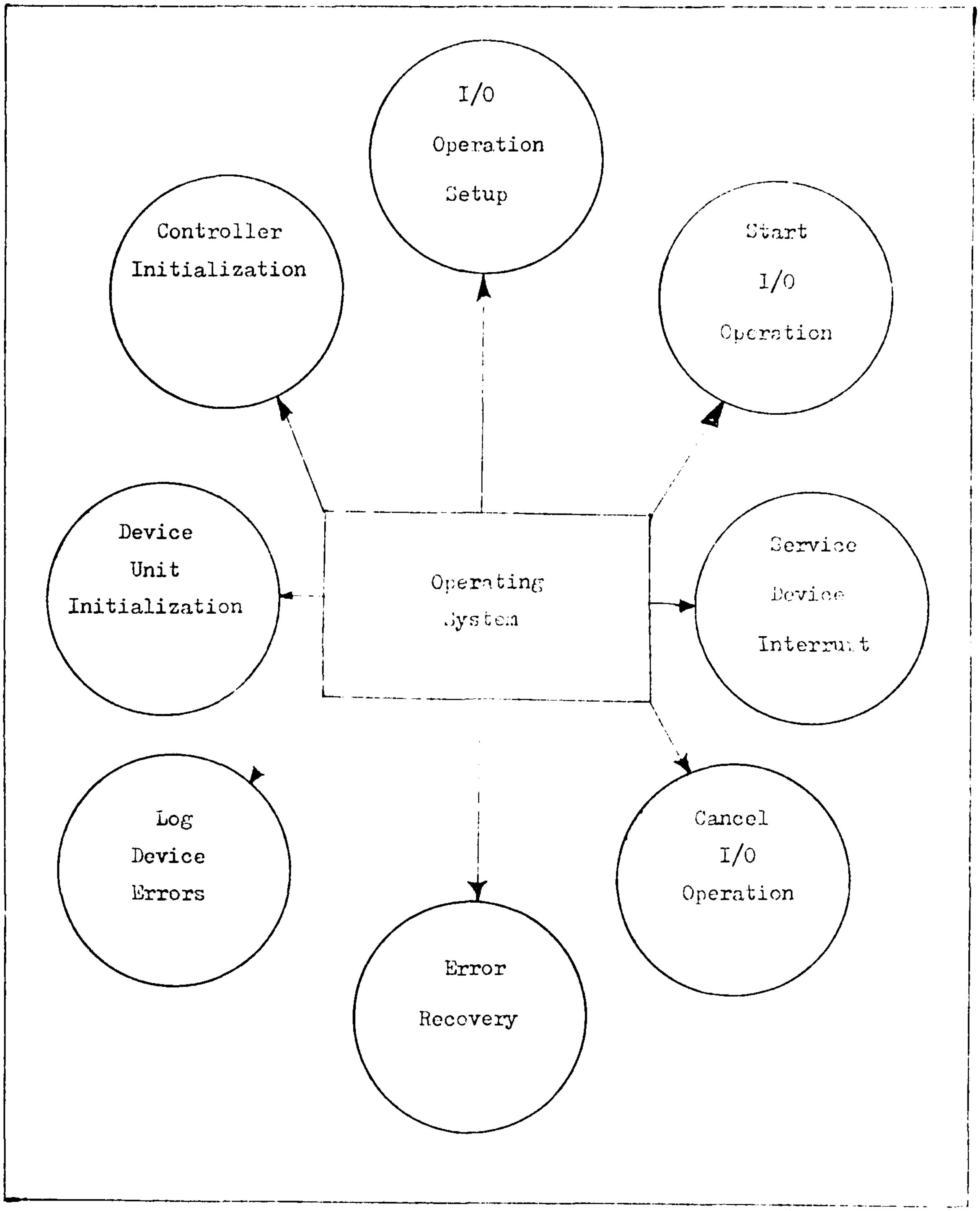
Interrupt Handling: Respond to hardware interrupts, read, and reset device registers; return status.

Error Recovery: Set up device registers for retry of an I/O operation; apply Error Correction Code (ECC) to disk data; return error status.

Error Logging: Write the contents of device registers and other data into an error buffer.

Cancel I/O Set up device registers to terminate I/O activity.

Device drivers need not contain all the subroutine types listed above. Every driver must include subroutines to handle I/O startup and interrupts. Figure 9 illustrates operating system interaction with I/O driver subroutines. (Ref. 22: 215-217)



(Ref. 22: 217)

Figure 9 Operating System Calls to Driver Subroutines

The hardware and software components of the VAX 11/780 provide the performance, reliability, and programming features often found only in much larger systems. The VAX 11/780 is a highly reliable system that is both flexible and extendable. In addition, the VAX 11/780 has some useful utilities and services that provice performance type information. These are:

- Display Utility
- Accounting.Dat File
- Operator's Log File
- Error Logger
- $Get Chan
- $ GET DEV
- $ INFO
- Show Status
- Show Process
- Show System

The display utility is perhaps the most useful of these services. The type of information collected is:

- File system statistics
- I/O system activity
- Use of processor modes
- Page management statistics
- Nonpaged pool statistics
- Activity in the scheduler state queues
- Principal users of CPU time
- System process activity

Each time the system is booted, it starts accumulating a new set of performance measurement statistics. The Display Utility Program provides a dynamic display of system performance measurement statistics on a VT-100 or VT-52 video display terminal. By typing appropriate digital control language commands, system users can obtain information about system activity. For more information on the Display Utility refer to the System Managers' Guide.

Another useful tool is the accounting.dat file. The VAX/VMS system creates and maintains, by itself, records on the use of system resources for accounting purposes. These records are kept in an accounting log file. The system updates the accounting log file when one of the following conditions is met:

- An interactive process terminates
- A batch process terminates
- A subprocess or a detached process terminates
- A printing job is completed
- A login failure occurs
- A user sends a message to the accounting log file by use of the Send Message to Accounting Manager system service.

By using the detailed accounting log records provided by the system, the system manager or a system programmer can establish programs for reporting on the use of system resources. DEC does not provide an accounting package, it is up to the individual users to write and maintain their own. The accounting package in use at SEAFAC was obtained through DECUS. The system service manual provides more detail on the accounting.dat file.

The operator's log file is yet another useful tool for obtaining computer performance information. The operator's log file is a system management tool that is useful in anticipating and preventing failures of both the hardware and the software. By regularly examining it, the manager can often detect tendencies or trends toward failures. This allows the system manager to take corrective action before such failure can occur.

The error logger is a job that runs continuously to log errors detected by both hardware and software. The errors include:

- Device errors
- Interrupt timeouts
- Interrupts received from nonexistant devices
- Memory, translation buffer, and check parity errors
- Datapath errors

The error logger writes all messages it receives into an error log file, noting vital system statistics at the time of the message. The error logger also notes benign events when they occur, such as when volumes are mounted and dismounted and provides periodic time stamps indicating that no entries have occurred for a specified period of time. The error logger can accept messages from system operators at any time and from any programs priviledged to send messages to the error logger. This system also includes a utility called the error report generating utility program (SYE) that converts the information in the error log file into a text file that can be printed for later study.

For more information refer to the Systems Service Manual.

In addition to these sources, the VAX 11/780 also has several commands which the system manager can use when he wants specific information. These commands are show status, show process, show system, and get I/O channel info ($GET CHN) and get I/O device info ($GET DEV). The show status command displays the following information:

- Current time and date
- Elapsed CPU time used by the current process
- Number of page faults
- Open file count
- Buffered I/O count
- Direct I/O count
- Current working set size
- Current amount of physical memory occupied

The show process command displays information about the current process. This command displays the following information about the current process:

- Date and time the show process command is issued
- Device name of the current SYS $ Input device
- User name
- Process identification number
- Process name
- User identification code (UIC)

- Base execution priority
- Default device
- Default directory
- Devices allocated to the process and volumes mounted, if any

The show system command displays a list of processes in the system and information about the status of each. The response displays:

- Process identification
- Process name
- User identification
- Process state
- Current priority
- Direct I/O count*
- Elapsed CPU time*
- Number of page faults*
- Physical memory occupied*
- Process indicator

*This information is displayed only if the process is currently in the balance set; if the process is not in the balance set, those columns contain the message "swapped out".

The get channel command returns information about a device to which an I/O channel has been assigned. The I/O device returns information about an I/O device. This service allows a process to obtain information about a device to which the process has not assigned a channel.

Although these sources provide useful information, there is some information that cannot be obtained by these sources. Table III contains a list of items that cannot be obtained by the above sources.

Summary

The purpose of the information part of the computer performance evaluation management system is to provide the members of a CPE team with a means to look for and identify computer performance problems. The information presented in this chapter shows the kinds and types of detailed information that can be obtained about a Data Center by using the guidelines and procedures of the information part of the CPE management system. This part of the system explains factors about a Data Center that can cause performance problems. It explains what to look for and how, and provides the members of the CPE team to immediately start their job of measuring and evaluating the performance of the computer system.

TABLE III

Information Which Presently Cannot Be Obtained

Peripheral Allocation Times	-	Begin and end times peripherals are allocated to a job.
Peripheral Busy Time/File	-	Total time spent in actual data transfer plus access time or a device, by file, for the job.
Channel Busy Time/File	-	Busy time for each channel, by file, for the job.
Processor Time	-	Time spent in execution of operating system, instructions in support of a job, by processor.
Processor Support Time	-	Time spent in execution of operating system instructions in support of a job, by processor.
Memory Requested	-	Amount of memory requested by a job.
Memory Used	-	Amount of memory used by a job.
Device Busy Time	-	Total busy time for each device in specified time intervals (e.g. 10 min., 1 hr., etc.)
Device Storage Available	-	Amount of unused storage available for each device in specified time intervals (e.g. 10 min., 1 hr., etc.)
Processor Time Application	-	Total processor time spent in direct execution of job instructions in specified time intervals (e.g. 10 min., 1 hr., etc.)
Processor Time Support	-	Total processor time spent in execution of operating system instruction; in specified time intervals (e.g. 10 min., 1 hr.)

TABLE III (continued)

Processor Time Support/Job	-	Total processor time spent in execution of operating system instructions in support of a specified job; in specified time interval (e.g. 10 min., 1 hr., etc.)
Processor Idle Time	-	Total processor idle time in specified time intervals (e.g. 10 min., 1 hr., etc.)
Memory Available	-	Total memory available in the system for non-operating jobs.
Average Memory Used	-	Average memory used by all non-operating system jobs in specified time intervals (e.g. 10 min., 1 hr., etc.)
Average Memory Used by O-S	-	Average memory used by the operating system in specified time intervals (e.g. 10 min., 1 hr., etc.)

V. Recommendations

The recommendations presented in this section relate solely to the SEAFAC organization and pertain to the implementation of the computer performance evaluation management system developed by this investigation.

The primary recommendation presented to the system manager of SEAFAC was to implement the CPE management system and perform a full analysis of the organization (i.e. evaluate the organization, workload, and computer system hardware/software). Unfortunately, the system manager was unable to comply with this recommendation; the major drawback being the people needed to staff the CPE team. The system manager believes that the people within the organization, who could take on these responsibilities are presently working at their maximum and adding additional responsibilities on them would tend to divide their attentions and possibly result in delays to the present project. The next recommendation presented to the system manager of SEAFAC was to implement the CPE management system and conduct a partial analysis of the organization (i.e. focus on either the organization, the workload, or computer system hardware/software). This is the recommendation the system manager of SEAFAC complied with.

To conduct a partial analysis requires at least one person to be on the CPE team. For SEAFAC, this person will be the systems manager. Since the system manager has other duties and responsibilities, the time and effort he can spend performing the analysis will be limited.

Therefore, the system manager must review performance information that requires little time and effort.

The system manager of SEAFAC is fortunate for several reasons. First and foremost is that the organization is presently not experiencing any performance problems. Secondly, the computer system is presently more than capable of processing the exisiting workload and lastly, the computer system has many excellent utilities that provide computer performance information.

The system manager of SEAFAC will be conducting a partial analysis of the computer system hardware/software. Following is a list of recommendations that will assist the system manager when conducting the analysis; some of which have already been accomplished. Although these recommendations may seem trivial, one must remember they are being directed at one person who must perform this task with limited time and effort.

1. Establish procedures whereby the information from the existing accounting package is gathered and reviewed on a monthly basis.
2. Attempt to get developed a more extensive accounting package better suited to SEAFAC needs. Some of the information to be gathered would be that presented in Table IV. An excellent source for this development would be AFIT with a follow-up to this thesis.
3. Contact other VAX users and ask what performance measures or tools they use to evaluate their system and ask how

they were obtained. This can be extremely helpful and beneficial since some system managers may have discovered other tools or techniques for measuring the performance of the VAX 11/780.

4. Review more closely the information from the Digital Equipment Corporation User's Group (DECUS) meetings. Many problems and topics are discussed at these meetings and the proceedings are forwarded to all members. Getting in contact with a person who presented a topic of interest is very important since documentation on topics presented at these meetings is either extremely limited or non-existant.

5. Establish procedures whereby the operators log file is printed on a daily basis and review this file. This is useful in anticipating and preventing failure of hardware and software.

6. Review the I/O rates, page faults, users and top users parameters of the display utility during peak working periods of the day. This allows the system manager to study the impact of the workload on the computer system.

7. Depending on particular interest, review the SHOW STATUS, SHOW SYSTEM, and SHOW PROCESS commands. The information provided by these commands has previously been discussed.

8. Make computer performance a key issue and solicit the help of subordinates to identify problems.

9. Ask the vendor if there are other ways of obtaining computer performance information from the computer system which may be unpublished.

10. Establish a computer performance evaluation team and conduct a full analysis as soon as workload requirements permit.

VI. CONCLUSION

The objective of developing a computer performance evaluation management system was achieved by this thesis investigation. This system can be used by managers of data centers or installations to identify performance problems, provide information on computer performance, and to assist them in making long term plans for computer resources. This system also includes guidelines on how to establish a CPE team and samples of computer performance reports that the CPE team can generate. This system also contains factors about a data center that can cause performance problems, as well as methods for identifying them. This system was developed to provide data center managers with a means of measuring and evaluating the performance of their computer systems. The computer performance evaluation management system developed by this thesis provides this means and can be used at any data center or computer installation. The major drawback to this system is the people needed to staff the CPE team. Many data center managers, because of work requirements, will not want to assign CPE responsibilities to individuals within their organization. This is even more true if their organization has not experienced any computer performance problems. These organizations usually have computer performance measurement low on their list of priorities and it takes a higher priority, only when problems occur.

Computer Performance Measurement and Evaluation is definitely not an easy task. It requires a lot of knowledge, thought, and

AD-A115 538   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 5/1
A MANAGEMENT SYSTEM FOR COMPUTER PERFORMANCE EVALUATION.(U)
DEC 81 H K BIRCH
UNCLASSIFIED AFIT/GCS/EE/81D-1 NL

2 OF 2
AD A115538

END
DATE
FILMED
7-82
DTIC

hard work to plan and conduct such a program. As systems become more complex and more corporations and businesses depend on computers for support, the importance of performance measurement increases drastically. The result of this is that system managers and installation managers will be faced with more difficult and demanding questions concerning computer system performance. To answer these questions, the system manager or installation manager will need information that can only be provided by a system that continuously gathers and presents computer performance information.

Although this thesis explains how to start and continue a performance effort, it should not be looked upon as being dormant. It should be viewed as a dynamic system; a system that can be added onto and changed as new techniques and tools in computer performance measurement are developed.

Computer performance measurement is and will continue to be an important asset of any computer installation. It should not be viewed as something that will only be accomplished when problems occur. Instead, computer performance measurement and evaluation should begin the moment the computer system first becomes operational for the organization and continue throughout the life of the system.

Bibliography

1. Bell, T.E., et. al., Computer Performance Analysis: Framework and Initial Phases for a Performance Effort. Rand Technical Report R-549-1-PR, Rand Corporation, California, November 1972.

2. Bell, T.E., "Performance Determination - The Selection of Tools, If Any," National Computer Conference, 1973.

3. Buzen, J.P., "Fundamental Laws of Computer System Performance," Proceedings of the International Symposium on Computer Performance Modeling, Measurement, and Evaluation, 1976.

4. Ferrai, Domenico. Computer Systems Performance Evaluation. Englewood Cliffs, New Jersey: Prentice Hall, Inc. 1978.

5. Hanson, Jon G., Design and Implementation of USAF Avionics Integration Support Facilities, MS Thesis, Wright Patterson Air Force Base, Ohio: School of Engineering, AFIT, December 1981.

6. Kiviat, Phillip J. and Morris, Michael F., "Getting Started in Computer Performance Evaluation," CMG Transactions, Number 10 December 1975.

7. Lipovick, G. Jan, The Performance Imperative, Lecture Notes from DODCI Presentation, Testdata Corporation, November 1975.

8. Madnick, Stuart E. and Donovan, John J., Operating Systems. New York: McGraw Hill Book Company, 1974.

9. Nutt, Gary J., "Computer System Monitoring Techniques," National Technical Information Service, PB-233372, February 1973.

10. Nutt, Gary J., "Tutorial: Computer System Monitors," IEEE, Vol. 8, No. 11, November 1975.

11. Sreenivasen, K., On the Use of Accounting Data in the Evaluation of Computer System Performance, Mitre Tech Report, MTP-149, January 1974.

12. Stevens, David F., "How to Improve Your Performance Through Obfuscatory Measurement," AFIPS Conference Proceedings, Volume 47, National Computer Conference, California, June 5-8,1978.

13. Svobodovia, Liba, *Computer Performance Measurement and Evaluating Methods, Analysis and Applications*. New York: American Elesview Publishing Company, Inc., 1976.

14. Warner, C. Dudley, *Measurement Tools*. Testdata Systems Corporation, Santa Clara, California, August 1975.

15. Weinberg, Victor, *Structured Analysis*, New York, Yourdin Press 1979.

16. *Management Guidance for Developing and Installing an ADP Performance Management Program*. GS-77-5, General Service Administration. Washington, D.C. 1977.

17. "Performance Measurement in EDP." The EDP Audit, Control, and Security Newsletter (EDPACS) May 1974.

18. *Research and Development Needs for Computer Performance Evaluation*. Project No. 572C. The Mitre Corporation, Bedford Massachusetts, 1976.

19. *Record Management Services, Volume 7*, Reference Manual. Digital Equipment Corporation, Maynard, Massachusetts, 1978.

20. *System Manager's Guide, Volume 10*, Product Manual, Maynard Massachusetts, Digital Equipment Corporation, 1978.

21. *Systems Services and I/O, Volume 4*, Reference Manual. Digital Equipment Corporation, Maynard, Massachusetts, 1978.

22. "Tools and Techniques for Improving the Efficiency of Federal Automatic Data Processing Operations." *Report to the Congress* by the Comptroller General of the United States, 1974.

23. "Technical Summary: VAX 11/780," Digital Equipment Corporation, Maynard, Massachusetts, 1978.

24. *Total Software Requirements*. (DMF Ltr. March 18, 1977).

25. "*VAX 11 Software Handbook*." Reference Manual, Digital Equipment Corporation, Maynard, Massachusetts 1978.

Reference

1. Bell, T.E., Computer Performance Management through Control Limits. TRW-SS-76-01, Redondo Beach, California, TRW, January 1976.

2. Cockran, J.S., and Crockett, E.D., "Interpreting the Results of a Hardware System's Monitor," Spring Joint Computer Conference 1971.

3. Dodson, Philip. Control Data Corporation Cyber 70: Procedures for Performance Evaluation. MS Thesis, Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, May 1974 (AD-785129)

4. Drummond, Mansfield E., Evaluation and Measurement Techniques for Digital Computer Systems. Englewood Cliffs: Prentice Hall, Inc. 1973.

5. Dunlavey, Richard R., An Introduction to Computer Performance Measurement with Hardware Monitors, Testadata Systems Corporation, 1972.

6. Joslin, Edware O., "Nine Alternatives to a New Computer," Journal of Systems Management, Volume 25, No. 11, November 1974, Pg. 28-41.

7. Kelly, John C., "The Statistical Nature of Computer Performance Evaluation." Proceedings of Computer Performance Evaluation Users Group, September 23-26, 1975.

8. Kobayashi, H., Modeling and Analysis: An Introduction to System Performance Evaluation Methodology, Philippines, Addison-Wesley Publishing Company, Inc. 1978.

9. Kuck, D.J., and Kumar B., "A System Model for Computer Performance Evaluation," Proceedings of the International Symposium on Computer Performance Modeling, Measurement, and Evaluation. Cambridge, Massachusetts, March 23-31, 1976, pp. 187-199.

10. Schaeffer, Howard, Data Center Operations, Englewood Cliffs, Prentice-Hall, Inc. 1981.

11. Svobodovia, Liba, "Computer System Measurability," IEEE Computer Society, June 1976, pp. 9-17.

12. Weinberg, Gerald M., "The Psychology of Improved Programming Performance," Datamation, 1972.

APPENDIX A

Identifying Problems With the Organization

Appendix A contains a list of questions that should assist the analyst identify these kinds of problems.

1. Is the organization properly manned to fulfill its mission?
2. Are managers attentive to problems identified by subordinates?
3. Does the organization have close contact with the users, and is it concerned if products are late?
4. Does the organization have a training program in effect to educate newly assigned personnel of the organization mission?
5. Does the organization measure performance and how?
6. Does the organization have standard procedures for solving problems or are they haphazard?
7. What sort of procedures do managers use to insure them that all work was finished and no problems occurred?
8. What governs the operational hours of the computer system, workload or manning?
9. Is the organization structured properly to fulfill its mission? (e.g. does it have too many operators and not enough programmers or vice versa)
10. What is the general feeling by operators and programmers toward the computer system? (Do they treat it as a free good or are they concerned with performance?)

APPENDIX B

Understanding the Organization

Appendix B contains a list of questions that can be used by the analyst or system manager when developing an understanding of the organization.

1. Determine situations or circumstances that provoked the CPE effort.
2. Determine the number of personnel the organization has to include number of programmers, operators, system analyst and maintenance technicians as well as the different levels of management.
4. Obtain information on how it provides service to the customers.
5. Determine what computer system the organization uses.
6. Obtain information on the hours worked by programmers, operators, system analyst, and maintenance technicians.
7. Obtain information on the number of shifts and personnel requirements for each shift.
8. Determine the organization's operational hours or hours the organization is open for business.
9. Obtain information on management directives, operational procedures, policies, etc., used to govern the organization.
10. Obtain information on policies and procedures used to govern computer usage.
11. Determine the structure of the organization.

12. Determine the organization's operational objectives.

13. Determine the computer center's position with respect to the organization that it serves.

14. Obtain information on the hours per day and week the computer system is operational.

15. Determine if the computer center is run under a closed or open job or if theirs is a mixture.

16. Determine when and if the computer system is shut down and for what purposes.

17. Determine if the organization uses priorities or schedules in processing and why.

18. Obtain information on critical jobs; such as payrolls the computer center may process.

19. Conduct discussions with operators, programmers, system analyst, and maintenance technicians to ascertain information about performance problems they may have encountered.

20. Determine what kinds of jobs the computer center processes and how these jobs are entered into the computer system.

21. Conduct discussions with the installation manager and determine the kinds of information he needs to make computer performance decisions.

22. Determine how and when he would like this information presented.

23. Determine if the organization is involved in any measurement or evaluation activities such as gathering accounting data or using hardware/software monitors.

24. If so, determine the information provided and when and how this information is presented to management.

25. Constantly look for policies, procedures, directives, etc. either carried out or required by the organization that could cause poor performance. (Ref. 1: 11)

APPENDIX C

Identifying Problems With the Workload

Appendix C contains a list of questions that should assist the analyst identify problems with the workload.

1. Does the computer sit idle when jobs could be processing?
2. Does the organization need to establish another shift or hire more people to insure all processing gets finished?
3. Do schedules need to be changed to distribute processing requirements evenly?
4. Are the users running only jobs they need or are they running nice-to-have jobs, which require extra processing?
5. Do priorities need to be changed to allow for a more evenly processing time for all jobs?
6. Do procedures need to be changed that govern entering jobs into the system? (i.e. wait until you have at least ten jobs before you enter them, or enter a job whenever it arrives)
7. Does equipment in the computer room need relocating in order to provide a more efficient working environment? (i.e. operators require little time to find and load disk pack and tapes)
8. Do closer controls need to be established to show when a job arrives, completes processing and is returned to the customer? (This prevents customers from complaining about slow turnaround since you can tell them exactly when their job was returned)
9. Are all online users aware of the organizations' operational

hours and are they kept informed of all changes?

10. Do procedures need to be changed that govern customers picking up their products? (i.e. every morning or as soon as they are finished)

APPENDIX D

Understanding the Workload

Following is a list of questions that can be used by the analyst or system manager when developing an understanding of the workload.

1. Determine the kinds and types of jobs the installation processes.
2. Determine the job classifications and statistical groupings of jobs that are run on the computer system to include user and system jobs.
3. Determine the number of system and user jobs the computer processes.
4. Obtain information on scheduling policies and determine when and how jobs are scheduled.
5. Determine the approximate processing times of all jobs.
6. Determine the time the computer is dedicated to actual processing.
7. Determine if the computer is capable of processing all jobs or if backlogs exist.
8. Determine if the backlog occurs regularly or sporadic.
9. Determine what specific projects, programs and personnel use or request the services of the computer center.
10. Determine if priorities exist and how they are assigned.
11. Determine if the computer ever sits idle.
12. Talk with operators and ascertain if they have experienced any problems with the workload.
13. Obtain information on the largest and smallest user to include approximate number of jobs submitted by each.

14. Determine the maximum number of users.

15. Obtain general information on the jobs they process.

16. Determine if any deadlines or unusual processing is required by any of these users.

17. Talk with users and ascertain if they have experienced any problems. with the computer, organization, or receiving products on time.

18. Determine how the majority of jobs enter the system, e.g. by cards or terminal.

19. If there is a mix, determine the percentage of each.

20. Determine if bottlenecks exist and obtain information as to when and why.

21. Determine what time the computer is the busiest and the least busy.

22. Determine which jobs require hard copies and identify user jobs that must be run twice to fulfill hard copy requirements.

23. Obtain information on any accounting data the organization may gather.

24. Determine the procedures used to manage jobs that fail to execute and must be rerun.

25. Look for areas of the workload that could cause poor computer performance, such as one user scheduling all jobs to be run every Monday. **(Ref. 1: 12)**

APPENDIX E

Understanding the Hardware

Following is a list of questions that can be used by the analyst or system manager when developing a basic understanding of the computer hardware. The reference manuals provided by the vendor will provide more knowledge in these areas for those wishing a more indepth understanding or those needing more knowledge to solve a particular problem.

1. Determine the make and model of computer system used by the organization.
2. Determine the amount of memory, both actual and virtual that the computer has.
3. Determine the multiprogramming level.
4. Find out how many processors the computer system has.
5. Find out the speed of the processors.
6. Determine the capacity of the storage devices.
7. Find out how many I/O channels are connected to the terminal.
8. Determine the kinds of I/O that can be performed along with the size of data than can be transferred.
9. Determine the speed and capacity of input and output devices such as card readers and line printers.
10. Find out how many terminals can be and are hooked up to the computer system. (Ref. 1: 13)

APPENDIX F

Identifying Problems with Computer Hardware

Appendix F contains a list of questions that should assist the analyst identify problems with the hardware.

1. Does the computer system have sufficient memory to handle job requirements?
2. Does the computer have sufficient storage space?
3. For organizations that require lots of printing, does the printer have sufficient speed or is a faster printer required?
4. Does the computer have sufficient I/O capability or is another faster I/O controller or channel needed?
5. Is the hardware suited for the workload?
6. Does the organization need to invest in more equipment, for example, a tape drive to be used to store files infrequently used by the customers? (This will save disk space.)
7. Does the computer system frequently break down?
8. Do maintenance technicians arrive quickly when problems occur or is there a delay?
9. Is the computer system housed in a controlled environment?
10. Does the organization have a backup capability in cases of excessive computer downtime?

APPENDIX G

Identifying Problems with O-S

Appendix G contains a list of questions that should assist the analyst identify problems with software.

1. Does the memory manager allow more than one job in memory at a time?
2. Is the operating system taylored for the workload, in other words, do the majority of jobs require excessive CPU time but the operating system was developed for jobs requiring lots of I/O?
3. Does the process scheduler select jobs on a first come, first serve basis, or does it select them on a priority; if it is a priority, could this delay other jobs?
4. Are memory partitions static but memory requirements variable resulting in waste of memory?
5. Does the computer system have virtual memory, and if so, are there problems with page swapping?

APPENDIX H

Understanding the Operating System

Appendix H contains a list of questions that can be used by the analyst or system manager when developing an understanding of the operating system.

1. Determine how the job scheduler and process scheduler work.
2. Determine how memory is allocated and deallocated.
3. Determine where a job goes when it first enters the system.
4. Find out what queues a job can enter and determine the requirements needed to leave these queues.
5. Determine how long a job is allowed to execute.
6. Find out where a job goes after execution.
7. Determine how the operating system handles I/O.
8. Determine how the operating system software allocates and releases storage.
9. Determine how it controls input and output devices and other resources of the computer system.
10. Determine what parameters the user can adapt to their environment.
11. Determine what version of operating system the organization is using.

APPENDIX I

Program Manager's Responsibility

The project manager's ultimate responsiblity is twofold: to the managers that receive the team's recommendations and to the individual or group who will implement the team's recommendations. For lack of a better term, these two groups will be referred to as the "customer" in the following list.

1. Understanding the customer's problem and translating it to the analyst staff and to management.
2. Knowing what is needed to "solve" the problem.
3. Knowing how a model or study can be used to fill the customer's needs. This should be summarized in a written statement of work.
4. Insuring that clear and consistent specification for a model and/or operating plans for a study are produced.
5. Judging the appropriateness of the model or study plan for the needs of the project.
6. Insuring that the planned approach to the model or study effort is logical and realizable.
7. Understanding the details of the technical approach.
8. Insuring that all essential tasks are included.
9. Insuring that no unnecessary tasks are included.
10. Knowing the use to which the output from each task will be put.

11. Judging whether or not the approach selected for each task is the best way to achieve the output.

12. Knowing what resources are required for each task.

13. Thoroughly understanding the kinds of technical and management abilities that will be required in a project.

14. Determining what must be provided by all parties involved (e.g. customer, contractors, other participants)

15. Insuring that commitments of all resources made to the project are honored.

16. Determine the quantity and pattern of management required in a project.

17. Insuring that schedules are met.

18. Periodically reviewing the adequacy of personnel skills, quantity of personnel, facilities, equipment, and information.

19. Following established quality control procedures and when necessary, establishing additional project-related procedures.

20. Establishing measures to prevent malperformance.

21. Insuring that malperformance can be detected.

22. Exercising positive cost control.

23. Insuring that all ideas are explored and exploited.

24. Reporting project status (cost and technical performance) on schedule, each month.

25. Delivering fully documented project reports to the customer.

26. Preparing and presenting oral project briefings to the customer.

27. Bringing problems to the attention of an appropriate manger once he has determined that higher-level assistance is necessary.

APPENDIX J

Tools and Techniques For Use in Computer Performance Evaluation

The following is a list of the main tools and techniques used by computer performance analyst when measuring and evaluating the performance of a computer system. Included in this list is a description of each of these items and the advantages/disadvantages of each.

- Personal Inspection
- Accounting Systems
- Hardware Monitors
- Software Monitors
- Benchmark
- Models

Personal Inspection

Personal inspection can imply an uninspired glance at the machine room. This sort of activity often leads to beliefs about an installation based more on preconceived notions than on reality. This "tool" usually is employed in an "analysis" involving occasional glances at a machine room when the observer sees precisely what he expected to see (whether it is true or not, and often even in the face of significant, contrary evidence). Since the observer may only glance at the machine room for a few minutes two or three times per day, his sample of the day's operation is very incomplete. This type of performance analysis, although common, is without redeeming social value and will not be considered further. Other types of personal inspection are more valuable for performance anlysis. An example of another type of personal inspection follows.

Each time a piece of unit record equipment processes a record, it emits a sound. The performance analyst can use this sound to roughly estimate activity and judge the occurrence of certain system-wide problems. For example, a multiprogrammed system may be experiencing disk contention in attempting to print spooled records. Quite often this problem manifests itself in strongly synchronized printing from several printers on a large system. As the disk head moves from track to track, first one, then another printer operates. When one printer completes output for its job, the other printer(s) begins operating at a sharply increased rate.

Multiple, rapidly spinning tapes and extremely active disk heads can, in some environments, indicate severe trouble. In other environments (where loads should be causing this kind of behavior), they may indicate a smooth running system. Unfortunately, most installations fall somewhere between these two extremes, leaving analysts and managers with an amorphous feeling of unease.

The clues from personal inspection can be valuable, but an experienced eye, accompanied with an equally experienced ear, is often necessary to make sense from the raw environment. Fortunately, other alternatives are available. (Ref. 2: 31,32)

Accounting Systems

Accounting systems aggregate computer usage by task, job, or other unit of user-directed work. Although accounting data can be deceptive, analysts can determine the actual data collection methods used and perform analysis based on a good understanding of potential errors. Accounting data also has some distinct advantages for analysis. They are usually quite complete because they are retained for historical purposes and changes in collection methods are well documented so that users can examine them for correctness. The data are collected about the system's work and organized in precisely the correct way to facilitate workload control - by requests for computer work (by job).

For most analysts, accounting data have the advantage of immediate availability so analysts,can begin without delays for acquisition of a tool; however, immediate data availability does not necessarily imply immediate useability. Accounting systems are commonly very extensive, so analysts are often overwhelmed with the quantity of items collected and the number of incidents of each item. All these data are usually placed in poorly formatted records on a file along with irrevelant or redundant data. The data conditioning problem may; therefore, be a major hurdle for successful analysis. Inadequate documentation of the details of data collection by manufacturers and inadequacies in the data collection (leading to variability in addition to significant bias) can confuse any analyst results unless the analyst is very careful. (Ref. 2: 32)

Hardware Monitors

A hardware monitor is normally a free-standing device which obtains signals from a computer system under study through high-impedence probes attached directly to the computer's circuitry. The signals can usually be passed through logic patchboards to do logical AND's, OR's, and so on, enabling the analyst to obtain signals when certain arbitrary, complex relationships exist. The signals are then fed to counters or timers. By the use of logic circuits (i.e., AND or GATES), it is possible to determine when specific hardware components are active, idle, or used concurrently. For example, an analyst with a hardware monitor could determine (1) the portion of CPU time spent performing supervisory functions while only one channel/controller is active, or (2) the number of times a channel becomes active during a certain period. Because hardware monitors can sense nearly any binary signal (within reason) they can be used with a variety of operating systems, and even with machines built by different manufacturers. (Ref. 2: 32)

Almost all hardware monitors need the computational power of a computer, at least to reduce the collected data. An advantage of hardware monitors is that their interference with the computer system is very minimal or none. Their disadvantage is that the installation generally requires great expertise and a thorough knowledge of the measured system and their users have to be carefully trained.

Some examples of hardware monitors are the Dynaprobe 7900, 8000, and Dyan-myte, all from COMRESS and the System 1000 from Testdata. (Ref. 18: 11) Currently these are the only two vendors in the field:

Comress Incorporated
Two Research Court
Rockville, Maryland 20850

Testdata Systems Corporation
7900 Westpark Drive
Mclean, Virginia 22101

Before hooking up probes; however, it is a good idea to discuss the project with the engineers who maintain your hardware. They are apt to be a little nervous about the monitoring process because they are afraid it might cause problems with the hardware. For the most part, these fears are unfounded. It does not hurt to have the engineers on your side; however, since they can give invaluable aid in locating probe points and suggestions on additional ones.

Hardware monitors may be purchased within the range of $4,000 to $100,000. The average device likely to offer desirable capabilities would cost about $35,000. It may be possible; however, to rent the hardware monitor for a short term or a one-shot project. (Ref. 17: VI-60)

Software Monitors

Software monitoring tools are defined as those consisting of instructions which are added to a hardware-software system in order to gather data related to its performance. This means that they can have access to the tables that operating systems maintain and thereby collect data that are more familiar to the typical performance analyst. Such a device can provide statistics on I/O device utilization, main storage usage, I/O wait time, idle time, and CPU time, etc. Usually the operating system must be altered in some way to collect the statistics. The fact that these additional instructions must be executed by the system being measured causes interference with the system. A five percent overhead factor can be anticipated; however, it can range as high as 20% and occasionally may be even worse. (Ref. 17: VI-60) The amount of interference produced depends on the frequency of the events to be detected and on the operations performed by the tool at the occurrence of each event.

A software monitor can be implemented in different languages, but for efficiency reasons and because of the need to reach the hardware levels, software monitors are generally implemented in a machine language. The main disadvantage of a software monitor is that it can detect only less frequent events. Thus, hardware tools may be used to verify the accuracy of certain software tools. The main advantage of a software monitor is its ease of installation; (i.e. no probes) however, experienced personnel are often needed for this operation. Additionally, some training is also necessary for using them and for interpreting their outputs. (Ref. 2: 31)

Several types of software monitors are available. Boole and Babbage have developed a problem program monitor (PPE - Problem Program Monitor) and a configuration analyzer (CUE - Configuration

Utilization Efficiency). (Ref. 18:11) These packages generally cost about $10,000 - $15,000 for outright purchases. Lending vendors are:

Boole & Babbage, Inc.
850 Stewart Drive
Sunnyvale, California 94086

Comress Incorporated
Two Research Court
Rockville, Maryland 20850

Information Research Associates
2317 Longview Terrace
Austin, Texas 78705

Some vendors have their own hardware/software monitors. IBM, for example has both a hardware and a software monitor. They are not sold but are used in support of their marketing effort to sell a new system or upgrade an existing configuration. Their personnel will do the monitoring, analyze the output, and provide you with a report. You are not allowed to become involved in reviewing the data or evaluating results. While the approach requires little work on your part and is provided free, there are no known cases where this technique has resulted in a recommendation to reduce hardware.

Other major computer manufacturers have software monitors for internal use but it requires arm-twisting to get them. There are also a number of software monitors of the homegrown variety floating about, although these are not commercially available. The user group of your manufacturer is a good source for this information. Table IV is a comparison of hardware/software monitors. (Ref. 17: VI 60-61)

TABLE IV

Comparison of Monitors

	Hardware Monitors	Software Monitors
Data Collection Method	Either event or time driven at your option.	Time driven
Hardware Dependency	Almost None	Restricted to IBM 360's and 370's
Overhead	None	Some
Accuracy	Accurate	Some distortion due to overhead.
Flexibility	Good	Limited. Can only perform functions included within its program.
Ease of Use	Poor	Good
Cost	$4,000 - $100,000	$300 - $20,000
Useful Life	Unlimited	Relatively short, like any program
Training Needed	Extensive	Slight
Advantages	Able to measure overall system activity. Easily modified to change or revise performance evaluation plan. Does not require software modifications.	Provides detail about an individual application program. Can concentrate on one program in a multiprogramming system. Able to analyze activity within the operating system.
Disadvantages	Is not able to pinpoint problems in applications. Difficult to focus on a particular problem in multiprogramming. Requires extensive experience to perform analysis. Requires knowledge of and access to hardware internal operations.	Does not have the ability to measure its own overhead. Cannot detect problems related to file organization. Cannot vary testing testing parameters while the test is is being run. Analysis output is limited to that specified by the vendor of the monitor.

(Ref. 17: VI-61)

Benchmark

Most analysis of computer system performance rely on either benchmarks or probablistic models. Benchmarks, which may consist of real programs, synthetic programs, or trace driven simulations are most useful when it is necessary to determine system behavior under a precisely specified workload. Benchmark results: however, can be surprisingly sensitive to the nature of the workload that the system is assumed to be processing and slight changes in the workload definition may sometimes produce significant different conclusions.

Table V illustrates the crux of the problem with a highly simplified example. Suppose that an analyst is comparing "round robin" (RR) with "first come, first served" (FCFS) scheduling algorithms for a central processor. Assume that the workload of three jobs; Job A with a duration of seven seconds; Job B with a duration of one second; and Job C with a duration of three seconds. The first line of Table V gives the average response time for the three jobs in the case where the order of arrival is "ABC" with all jobs arriving at approximately the same time. Note that the average response for FCFS is 18% higher than the average response time for RR with a quantum of two seconds. Thus, the benchmark results in line one indicate a definite preference for RR. In the second line of Table V, everything is the same except that the order of arrival is reversed. In this case average response time for RR is higher than average response time for FCFS. The second set of benchmarks results thus indicate a strong preference for

FCFS even though this benchmark contains the same set of jobs as the first. As a point of interest, the third line of Table V presents yet another arrival sequence in which the two scheduling algorithms produce exactly the same response time.

Table VI presents the completion times of individual jobs and the average response time (completion time) of the entire benchmark for each of the three workloads presented in Table V. It is assumed that the quantum size in the round robin scheduling algorithm is two seconds.

Table V

			Average Response Time	
First	Second	Third	FCFS	RR (Q-2)
A	B	C	8 2/3	7 1/3
C	B	A	6	6 2/3
B	A	C	6 2/3	6 2/3

Although the example in Table V is highly simplified, the dangers which it illustrates are very real. In particular, benchmark evaluations require specification of the system workload in complete detail. As a result, the analyst is often compelled to make subtle but critical decisions in areas where his knowledge is usually imprecise. This, in turn leads to confusing situations where seemingly equivalent benchmark studies produce different final conclusions.

TABLE VI

<u>Table A-1</u>

		FCFS	RR
Workload - ABC	A	7	11
	B	8	3
	C	11	8
	Average	8 2/3	7 1/3

<u>Table A-2</u>

		FCFS	RR
Workload - CBA	C	3	6
	B	4	3
	A	11	11
	Average	6	6 2/3

<u>Table A-3</u>

		FCFS	RR
Workload - BAC	B	1	1
	A	8	11
	C	11	8
	Average	6 2/3	6 2/3

(Ref. 4: 200-210)

Models

A model of a system is a representation of the system which consists of a certain amount of organized information about it and is built for the purpose of studying it. In the field of computer performance evaluation, there are basically two types of models; analytic and simulation. (Ref. 4:20)

Analytic models are mathematical expressions of the relation p - Sp(W) which is derived by analysis of the behavior of a systems functional model. The class of problems that is solvable with existing mathematical methods is very limited; many simplifying assumptions must be made even for the least complicated systems. Analytical models often focus on the problem of management of a specific system resource such as:

- CPU scheduling
- Scheduling of rotational I/O devices
- Management of hierarchical memories
- Channel scheduling
- Buffer storage allocation
- File organization (Ref. 21: 34)

There are many kinds and types of models but basically they can be grouped into three categories; structural, functional, and performance.

Structural Models

A structural model describes individual system components and their connections. Such a model provides a useful interface between the real system and a more abstract one. Structural models are most frequently represented by block diagrams. The level of detail in a block diagram can easily be varied since individual blocks can in turn be further laid down as self-contained block diagrams. Block diagrams generally show the paths of data flow as well as control flow but they do not specify the conditions governing this flow. Thus, block diagrams are suitable only for the first general level description of the system under study. (Ref. 13: 31)

Functional Model

A functional model describes how the system operates. A functional model defines the system such that the system can be analyzed mathematically or studied empirically. Functional models used in performance analysis can be divided into four groups:

- Flowchart Models
- Finite-state Models
- Parallel Nets
- Queuing Model

Flowchart models are suitable for studying program efficiency and execution time requirements. A flowchart model is a directed graph model where the nodes represent computational tasks and the arcs show the possible flow of control between tasks. Alternatively, the computational tasks may be viewed as being represented by the arcs, the nodes then being the branch and junction points in the modeled program or merely points separating different tasks. Given the execution time of the individual tasks and the probability of following the various individual arcs, the total execution time of the modeled program can be derived by a sequence of elementary transformations. Flowchart models of system components and users' programs can be used as building elements of a system model, tied together by a mechanism that stimulates system resource allocation and scheduling.

A finite-state model can be used for analysis of utilization of computer system resources. It too, can be represented by a directed graph; however, the nodes now represent the state of the system. The arcs represent the transitions between states.

Parallel nets are modifications of petri nets. Parallel nets are directed graphs made of two different types of nodes; transitions and places. Places with arcs directed into a transition are the conditions that must be satisfied concurrently if this transition is to occur. They are well suited for describing concurrent asynchronous operations that take place in a computer system. (Ref. 13: 32)

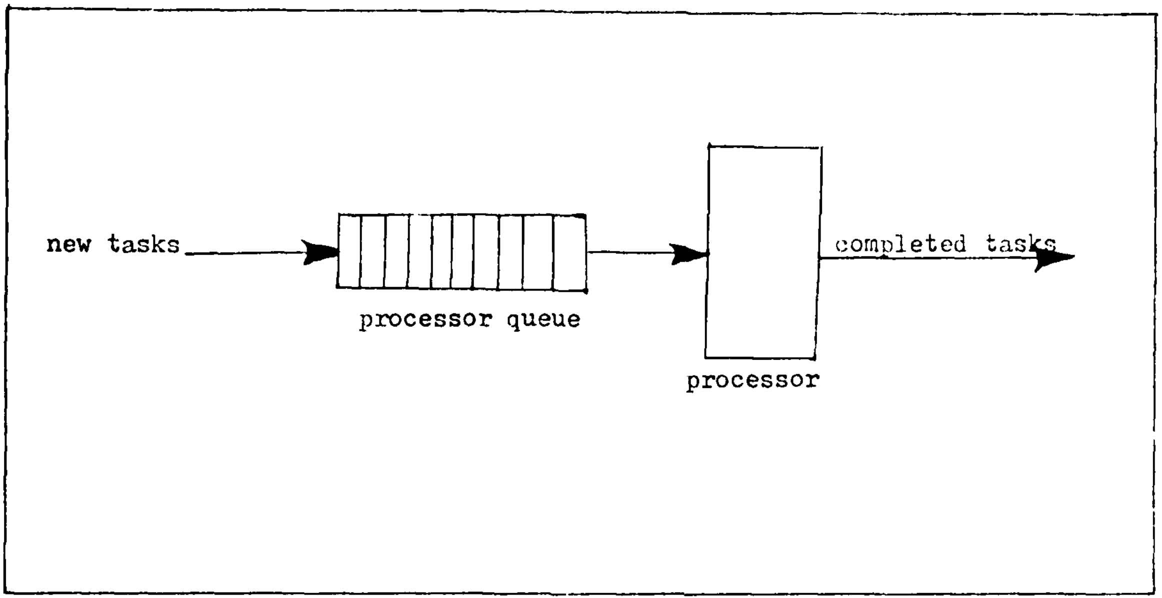
A queuing model is defined by its sources, its service centers, and their interconnections. The basic components of a queuing model are servers, queues, and sources. Servers are generally used to model the resources demanded by the jobs. The jobs are generated by sources or exist in the queuing model since its creation. Each server can only serve a limited maximum number of jobs at the same time. This is often called the number of channels of the server. Those jobs which find the server busy must wait in a queue until their turn comes. Each server has at least one queue, and the term service center is often used to indicate the complex consisting of a server and its queues. In some cases, a service center contains several servers, all of which process jobs from the same queue or queues. A job generally requests the attention of a server for a certain amount of time (called service time) and joins a service center at an instant called the arrival time or the job at the center. (Ref. 4: 178-179)

The simplest type of queuing model is the single-service center (or single server) model depicted in Figure 10. The service center consists of a single-channel server and of one queue with unbounded capacity. When a job has been completely processed by the server, it leaves the model. (Ref. 13: 35)

The most popular computer performance indices (response time, turnaround time, throughput rate, utilization factors) are usually easy to define; though not always easy to compute in a queuing model. Other easily definable indices are the waiting times in the queues and the queue lengths. Of course, component oriented indices such as utilization, waiting times, and queue lengths require, to be defineable, that the corresponding component be explicitly modeled in the network. (Ref. 4: 180-181)

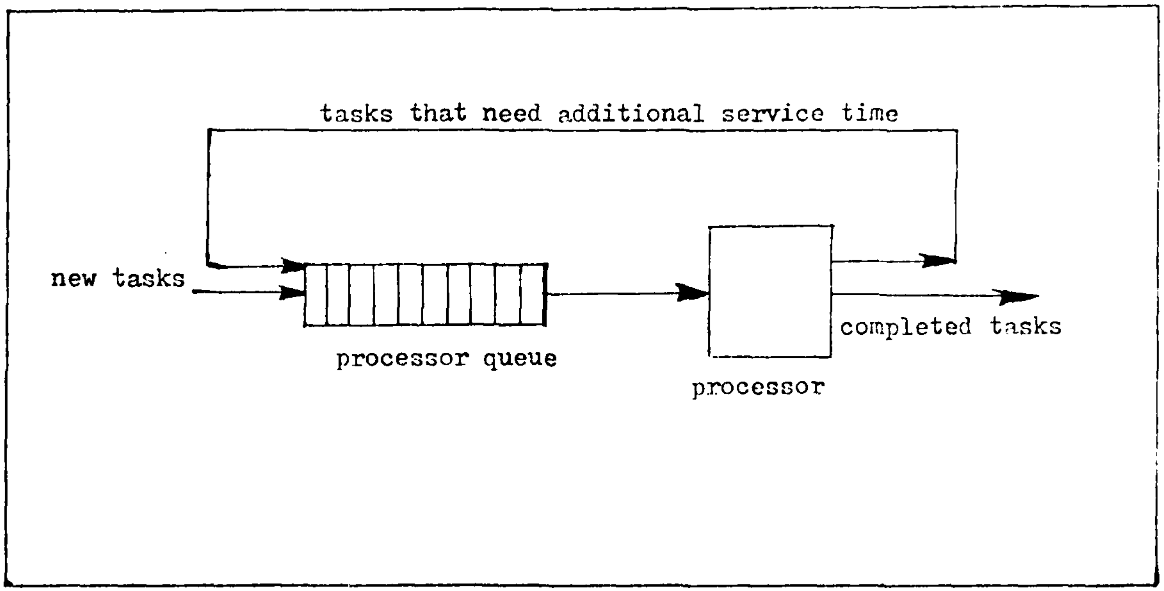
Queuing models are further classified according to the service discipline which is a rule that determines how the requests are processed. The simplest discipline is the first-come-first-served (FCFS) discipline where the requests are processed simply in the order of their arrival. More elaborate service disciplines were developed to increase system throughput and lower the total time a task spends in the system (turnaround time or response time). The round-robin (RR) discipline allocates one time quantun to a task at the head of the queue. If a task requires additional time after receiving its quantum, it is placed at the end of the queue. The model of a round-robin discipline is shown in Figure 11.

Queuing may occur for any system resource that can be used by several active jobs, but only one job at a time (CPU, channels, I/O controllers, disks and drums, memory blocks). A complete system can be modeled as a network of interfacing queues. Most of the queuing networks; however, are variations of the central server model that handles queuing for several different I/O processors.



(Ref. 13: 35)

Figure 10 Simplest Single-Server Model



(Ref. 13: 37)

Figure 11 Round-Robin Model

Queuing models emphasize the flow of jobs through the system. They also enable one to observe the state of the system and they are the most widely used models in computer performance analysis. (Ref. 13: 34-36)

Vita

Harry K. Birch was born on 29 November 1952 in Nassawadox, Virginia. He graduated from Chincoteague High School, Virginia in 1971 and attended Lenoir Community College in Kinston, North Carolina where he received an Associate Degree in Aviation Management and obtained a commerial pilot license. He entered the ROTC program at East Carolina University in Greenville, North Carolina where he received a Bachelor of Science degree in June of 1975 along with a commission in the USAF. He entered the Air Force on 1 September 1976 and the next three years were spent as a Data Processing Installation Manager at Myrtle Beach AFB, South Carolina. He entered the AFIT resident school in June 1980 and received the degree of Master of Science in Information Systems in December 1981.

Permanent Address: Box 162, Deep Hole Road
Chincoteague, Virginia 23336

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AFIT GCS/EE/81D-1

2. GOVT ACCESSION NO.
AD-A113 538

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A Management System for Computer Performance Evaluation

5. TYPE OF REPORT & PERIOD COVERED
Thesis

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Harry K. Birch, Captain

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Air Force Institute of Technology
AFIT-EN Wright Patterson AFB, Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Systems Engineering Avionics Facility
Aeronautical Systems Division, WPAFB, Ohio 45433

12. REPORT DATE
December 1981

13. NUMBER OF PAGES
121

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
None

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

15 APR 1982

18. SUPPLEMENTARY NOTES
Approved for Public Release; IAW AFR 190-17 Lynn E. Wolaver Dean for Research and Professional Development
~~Frederic C. Lynch, Major~~, USAF
Dir~~ector of Public Af~~fairs
Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Computer Performance Evaluation Management System, Computer Performance Evaluation Team, Performance Factors, Computer Performance Evaluation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
As computers and computer systems became more complex, the difficulty of measuring and evaluating the performances of these systems increases drastically. Many data center managers and computer system managers are incapable of dealing with the complex issues of computer performance

evaluation. To assist these managers deal with the complex issues of computer evaluation, a management system for computer performance evaluation was developed. This system is composed of three parts; information, people, and reports..

The information part of this system is a set of factors about the data center that can cause problems with computer performance and methods to identify these factors. This part also includes the data which can be gathered by various CPE tools and techniques used to solve these problems.

The people part of this system are members of a special CPE team. This team uses the information part of the system to identify problems and recommend solutions to management. The qualifications needed by the members is discussed along with administrative and reporting procedures.

The reports part of this system is the most meaningful part seen by the manager. The reports generated by the CPE team provide the manager with the means to measure and evaluate the performance of the computer system. The timeliness and accuracy of the reports lies with the CPE team. Since there are numerous reports that can be generated about a data center, only the major ones are discussed along with their meaning and usefullness.

This system also includes background information on computer performance analysis, as well as explanations and definitions of many of the tools and techniques used by CPE analyst.

ILMED

8